

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 4月17日
Date of Application:

出願番号 特願2003-112992
Application Number:
[ST. 10/C]: [JP 2003-112992]

出願人 富士通株式会社
Applicant(s):

2003年12月16日

特許庁長官
Commissioner,
Japan Patent Office

今井 康



出証番号 出証特2003-3104502

【書類名】 特許願

【整理番号】 0340363

【提出日】 平成15年 4月17日

【あて先】 特許庁長官殿

【国際特許分類】 G06C 7/09

【発明の名称】 データ保障装置、データ通信装置、およびデータ保障方法

【請求項の数】 10

【発明者】

 【住所又は居所】 神奈川県川崎市幸区堀川町 6 6 番地 2 富士通エルエス
 アイソリューション株式会社内

 【氏名】 副島 真智子

【特許出願人】

 【識別番号】 000005223

 【氏名又は名称】 富士通株式会社

【代理人】

 【識別番号】 100092152

 【弁理士】

 【氏名又は名称】 服部 毅巖

 【電話番号】 0426-45-6644

【手数料の表示】

 【予納台帳番号】 009874

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9705176

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ保障装置、データ通信装置、およびデータ保障方法

【特許請求の範囲】

【請求項 1】 データを保障するための処理を行うデータ保障装置において

処理対象データを取得するデータ取得回路と、

入力されたデータの暗号処理を行う暗号処理回路と、

前記データ取得回路に第 1 のバスを介して接続されると共に前記暗号処理回路に対して第 2 のバスを介して接続されており、前記データ取得回路が取得した前記処理対象データを前記第 1 のバス経由で取得して内蔵メモリに格納し、前記処理対象データを前記第 2 のバス経由で前記暗号処理回路に入力し、前記暗号処理回路から前記第 2 のバス経由で暗号処理実行後の処理結果データを取得するデータ入出力制御回路と、

を有することを特徴とするデータ保障装置。

【請求項 2】 前記データ入出力制御回路は、前記第 1 のバスのダイレクトメモリアクセスコントローラを有しており、前記処理対象データを前記データ取得回路からダイレクトメモリアクセス転送によって取得することを特徴とする請求項 1 記載のデータ保障装置。

【請求項 3】 前記暗号処理回路は、入力された処理対象データの容量を検知し、所定の容量に達したときに前記処理対象データに対する暗号処理を実行することを特徴とする請求項 1 記載のデータ保障装置。

【請求項 4】 前記暗号処理回路が複数設けられ、複数の前記データ入出力制御回路が各前記暗号処理回路に個別に対応づけて接続されており、複数の前記データ入出力制御回路が前記処理対象データを分割して取得し、対応する前記暗号処理回路に対して並列に入力することを特徴とする請求項 1 記載のデータ保障装置。

【請求項 5】 入力されたデータの認証処理を行う認証処理回路を更に有し

前記データ入出力制御回路は前記認証処理回路に対して第 3 のバスを介して接

続されており、暗号化対象である前記処理対象データを前記暗号処理に入力し、認証対象である前記処理対象データを前記認証処理回路に入力することを特徴とする請求項 1 記載のデータ保障装置。

【請求項 6】 データを保障するための処理を行うデータ保障装置において、

処理対象データを取得するデータ取得回路と、

入力されたデータの認証処理を行う認証処理回路と、

前記データ取得回路に第 1 のバスを介して接続されると共に前記認証処理回路に対して第 2 のバスを介して接続されており、前記データ取得回路が取得した前記処理対象データを前記第 1 のバス経由で取得して内蔵メモリに格納し、前記処理対象データを前記第 2 のバス経由で前記認証処理回路に入力するデータ入出力制御回路と、

を有することを特徴とするデータ保障装置。

【請求項 7】 保障されたデータをネットワークを介して送受信するデータ通信装置において、

送信データを生成するメイン CPU と、

入力されたデータを暗号化する暗号処理回路と、

入力されたデータを前記ネットワークを介して送信する通信回路と、

前記メイン CPU と前記通信回路に第 1 のバスを介して接続されると共に前記暗号処理回路に対して第 2 のバスを介して接続されており、前記メイン CPU が取得した前記送信データを前記第 1 のバス経由で取得して内蔵メモリに格納し、前記送信データを前記第 2 のバス経由で前記暗号処理回路に入力し、前記暗号処理回路から前記第 2 のバス経由で暗号化後の暗号データを取得し、前記通信回路に対して入力するデータ入出力制御回路と、

を有することを特徴とするデータ通信装置。

【請求項 8】 保障されたデータをネットワークを介して送受信するデータ通信装置において、

受信データを処理するメイン CPU と、

入力されたデータを復号する暗号処理回路と、

前記ネットワークを介して送られた受信データを取得する通信回路と、

前記メインCPUと前記通信回路に第1のバスを介して接続されると共に前記暗号処理回路に対して第2のバスを介して接続されており、前記通信回路が取得した前記受信データを前記第1のバス経由で取得して内蔵メモリに格納し、前記受信データを前記第2のバス経由で前記暗号処理回路に入力し、前記暗号処理回路から前記第2のバス経由で復号後の平文データを取得し、前記メインCPUに対して入力するデータ入出力制御回路と、

を有することを特徴とするデータ通信装置。

【請求項9】 データを保障するためのデータ保障方法において、

データ取得回路で取得した処理対象データを、データ入出力制御回路が第1のバス経由で取得して内蔵メモリに格納し、

前記データ入出力制御回路が、前記処理対象データを第2のバス経由で暗号処理回路に入力し、

前記暗号処理回路が前記処理対象データの暗号処理を行い、

暗号処理実行後の処理結果データを、前記暗号処理回路から前記データ入出力制御回路に渡す、

ことを特徴とするデータ保障方法。

【請求項10】 データを保障するためのデータ保障方法において、

データ取得回路で取得した処理対象データを、データ入出力制御回路が第1のバス経由で取得して内蔵メモリに格納し、

前記データ入出力制御回路が、前記処理対象データを第2のバス経由で認証処理回路に入力し、

前記認証処理回路が前記処理対象データの認証処理を行う、

ことを特徴とするデータ保障方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はセキュアな通信を行うためのデータ保障装置、データ通信装置、およびデータ保障方法に関し、特にハードウェア回路により暗号処理を行うことがで

きるデータ保障装置、データ通信装置、およびデータ保障方法に関する。

【0002】

【従来の技術】

インターネットの普及により、インターネットを経由した様々なデータ通信が行われる。このデータ通信には、個人情報や企業機密に該当するデータも含まれる。このような重要なデータは、暗号化等によって第三者による不正取得行為から守られる。

【0003】

暗号処理（暗号化や復号処理）は、解読の困難性を高めるほど暗号化のアルゴリズムが複雑になる。そのため、暗号処理や認証処理をソフトウェアで実行するシステムでは、CPUにかかる処理負担が過大となっていた。その結果、暗号処理や認証処理がシステム全体の処理能力に大きな影響を及ぼしていた。

【0004】

特に次世代インターネットプロトコルIP v 6 (Internet Protocol version 6) では、IP v 4 (Internet Protocol version 4)においてオプション機能であったIPsecを必携機能と定めている。IPsecは、TCP/IP (Transmission Control Protocol/Internet Protocol)にセキュリティ機能を加えたプロトコルである。IPsecを適用すれば、LAN (Local Area Network)、私的もしくは公共のWAN (Wide Area Network)、さらにはインターネット越しの通信の安全性を確保することができる。

【0005】

IPsecは大きく分けてIKE (Internet Key Exchange)、ESP (Encapsulating Security Payload)、AH (Authentication Header)という3つのプロトコルからなり、それら全てに暗号処理が適用される。

【0006】

IPsecによる暗号化通信は、まず鍵交換を含めたSA (Security Association)の合意をとることから始まる。IPsecでは、自動でSAの合意をとることが可能な鍵交換プロトコルとして、「IKE」を規定している。IKEを使うことで、SAの合意を自動的に行うことが可能になる。なお、鍵交換プロトコル

としてはほかにもいくつか存在するが、IPsecでは「ISAKMP/Oakley」という鍵交換プロトコルをもとにして作られたIKEが標準となっている。

【0007】

ところで、この鍵交換の段階で通信データの内容が第三者に漏れた場合、以後のIPsecによる暗号化通信は何ら意味をなさなくなる。またIPsecによる暗号化は、鍵交換が終了して初めて有効になるので、IKE自体にIPsecを使うわけにはいかない。そこでIKEはそれ自体で暗号化通信をサポートしている。

【0008】

IKE自体の暗号化通信のためにさらにIKE用の鍵交換手順が定められており、このためIKEは全体で2つの段階から構成されている。まずフェーズ1では、フェーズ2の段階で利用する暗号化アルゴリズムを決定するとともに、暗号鍵を生成する。暗号鍵が共有できた後は、これを使ってフェーズ2に進み、IKE限定の暗号化通信が可能になる。続いて、IPsecによる暗号化通信のためのネゴシエーションを始める。

【0009】

暗号化通信のためのネゴシエーションでは、順次、暗号化アルゴリズムの決定、暗号鍵の交換などIPsecによる通信に必要な各種情報がやり取りされる。その後、データの暗号化通信が行えるようになる。

【0010】

データの転送に利用するプロトコル「ESP」ネゴシエーションが終了した後、当事者同士で暗号化されたパケットによる通信が開始される。IPsecではパケットごとに暗号化がなされ、「ESP」と呼ばれる入れ物にパックされ送信される。IPsecでは、暗号化する対象部分によって、「トランスポートモード」と「トンネルモード」という2つの方法が提供されている。

【0011】

トランスポートモードでは、IPパケットで運ぶデータ部分のみを暗号化し、これにあて先などを指定したIPヘッダをつけて送信をする。一方、トンネルモードでは、ほかのホストからいったん受信したIPヘッダとデータ部分を合わせ

たものをまとめて暗号化したうえで、新たに IP ヘッダを再度つけ直して送信を行う。

【0012】

また、通信データの改竄を防止する技術として、認証処理がある。認証処理では、通信データの内容が正しいことを証明するために、認証データが生成される。

【0013】

認証データは、「完全性の確保」と「認証」の機能を担うものである。この中身は、「MAC (Message Authentication Code)」といわれるデータが入れられる。MAC は、通信内容とパスワードを合わせたものに対して、ハッシュ関数と呼ばれる計算方法による演算を施した計算結果である。ハッシュ関数により、任意の大きさのデータから、数十ビットから数百ビットの固定長データが生成される。ハッシュ関数は暗号処理に似た処理であり、多くの場合、暗号処理内の一機能として、ハッシュ関数の認証処理が含まれる。

【0014】

上記暗号処理、認証処理アルゴリズムの代表的なものに DES-CBC、3DES-CBC、MD5、SHA1、HMAC-MD5、HMAC-SHA1 が挙げられる。

送信側では、データとパスワードを合わせたものをメッセージ・ダイジェストにより処理したのち、結果を ESP 中の認証データとしてパケットに付け加える。データが無事に受信側に届いたら、受信側ではデータと自分の側でとっておいたパスワードを合わせたものを送信時と同じメッセージ・ダイジェストによって計算をする。得られた結果と受信データを比較して、この 2 つの間で相違がなければ、データが途中で改竄されることなく届いていることになる。

【0015】

完全性と認証のためのプロトコル「AH」は、「完全性の保証」と「認証」のための仕組みである。AH ではデータの暗号化は行わず、SPI、シーケンス番号、そして認証データのみをパックして通常の IP パケットの中に加えるようになっている。ESP だけで認証機能も果たすが、AH が定められている理由は、暗号化通信が使えないというケースのために、最低限「完全性の保証」と「認証

」を行うために提供されている。なお、データの認証方式の有名なものには、ハッシュ関数アルゴリズムである S H A 1 (Secure Hash Algorithm 1) などがある (非特許文献 1 参照)。

【 0 0 1 6 】

現在の I P s e c の主な用途に、インターネットを使った「V P N (Virtual Private Network)」がある。これは従来、専用線により実現されていた企業での本支店間の接続、あるいは L A N 間接続といったものをインターネット経由で行うためのものである。インターネットでは不特定多数に通信内容がさらされることになるため、送信するデータを守る仕組みが必要になる。そこで、V P N に I P s e c を使用する。すると、利用料金は専用線と比較してはるかに安価でありながら、専用線と同じような通信の秘匿性を実現することができる。

【 0 0 1 7 】

現時点で主たる I P s e c 採用製品もこうした V P N を対象とするものが多い。製品の形態としては専用の暗号化装置、ルータやファイアウォール製品の付加機能といったものである。こうした製品を各拠点でのインターネットのアクセス回線の入口に設置し、前述のトンネルモードの I P s e c を利用することで、拠点間のすべての通信を暗号化することが可能になる。また、最近では、市販の O S (Operating System) が標準で I P s e c をサポートしたこともあり、S O H O (Small Office/Home Office) 同士や家庭とオフィス間での暗号化通信も利用可能になりつつある。今後、インターネットが社会インフラとして定着するにつれ、セキュリティは個人や企業といったユーザとしての立場を問わず必要不可欠な技術となっている。そのため、I P s e c は大変将来性のある技術である。

【 0 0 1 8 】

暗号処理、ハッシュ関数による処理全般や、ある特定のハッシュ関数を使用して H M A C (Keyed-Hashing for Message Authentication Code) メッセージ認証関数を構成する一般的な方法として C P U によるソフトウェア処理がある。この場合、暗号処理は積和演算、ビット転置変換、排他的論理輪処理、ビットシフトを多用するために、C P U にとって演算負荷が重く、低パフォーマンスの C P U では処理に時間がかかり過ぎて正常なプロトコル処理を保障することも難しい

。一方、高パフォーマンスのCPUでは充分速い速度で処理できるが、価格、消費電力も高く、またシステムとして安定動作し難い。また、高いパフォーマンスを持つCPUは高価であり、低いパフォーマンスのCPUしか組み込むことのできない安価な製品には使うことができない。

【0019】

ソフトウェアによる暗号処理の速度はCPU占有率とCPUの処理能力に完全に依存する。そのため、速度も十分ではない上に他の処理と組み合わせた時の状況下ではより一層の処理速度の低下と処理の困難性が高まる。たとえば、IPsecを行う場合、低パフォーマンスのCPUでは処理に時間がかかり過ぎて正常なプロトコル処理を保障することも難しい。なお、高パフォーマンスのCPUでは充分速い速度で処理できるが、価格、消費電力も高く、またシステムとして安定動作し難い。そのため、特に高いパフォーマンスを持たないCPUを持つ組み込み向け製品にとってIPsec導入において実用的なパフォーマンスと価格に対して実現が難航していた。

【0020】

このように、従来の技術では暗号処理に時間がかかり、ストリーミング処理をユーザに快適な速度で実現できない。IPsecの鍵交換の中で用いられる暗号処理、認証処理、IPパケットの暗号化、認証のような大きなデータの処理には膨大な時間がかかっていた。

【0021】

また、高速化のために暗号処理専用LSI (Large Scale Integration)チップを使用することもできる。暗号処理回路と同様に、ハッシュ関数処理に関しても回路で実現できる（たとえば、特許文献1、特許文献2参照）。また、IPsecに使用するストリーム暗号装置も考えられている（たとえば、特許文献3参照）。

【0022】

このような専用回路をシステムに組み込むことで、ソフトウェアを用いた処理と比較するとパフォーマンスやコスト、消費電力は改善される。また、ソフトウェア演算のようなメモリコピーでメモリを余計に使用することも無い。

【 0 0 2 3 】

【特許文献 1】

特表平 1 1 - 5 0 0 2 4 1 号公報

【特許文献 2】

特開 2 0 0 1 - 1 7 5 6 0 5 号公報

【特許文献 3】

特開 2 0 0 3 - 3 2 2 4 4 号公報

【非特許文献 1】

Bruce Schneier "Applied Cryptography (Second Edition) ", 1996,
John Wiley & Sons, Inc P265~P278, P429~P459

【 0 0 2 4 】

【発明が解決しようとする課題】

しかし、従来の技術では専用回路へのデータ入出力には CPU が介在する。すなわち、このような専用回路は処理ブロック数毎に CPU が専用回路内部レジスタにライト、リード動作を行わなくてはならない。そのため、レジスタリードライトに CPU が煩わされることとなる。処理量が大きくなるにつれ、この点のパフォーマンスの悪さはより大きくなる。

【 0 0 2 5 】

また、従来の回路は内部に制御レジスタを設け、暗号化/復号化対象データのレジスタライトを終了した後に暗号演算開始ビットを ON にして処理を開始させるものである。そのため、ストリーミング処理に向かない。

【 0 0 2 6 】

このように、CPU が介在する処理が多いことにより、処理速度は CPU 占有率と CPU の処理能力に完全に依存する。そのため、専用回路の能力を十分に発揮できない上に、他の処理と組み合わせた時の状況下ではより一層の処理速度の低下や処理の困難性が高まると考えられる。

【 0 0 2 7 】

以上の理由により、安価に製造可能な回路で快適な通信速度を実現可能な専用回路とシステムを実現する必要がある。

本発明はこのような点に鑑みてなされたものであり、CPUの処理性能に拘わらず暗号処理を高速に行うことができるデータ保障装置、データ通信装置、およびデータ保障方法を提供することを目的とする。

【0028】

【課題を解決するための手段】

本発明では上記課題を解決するために、図1に示すようなデータ保障装置1が提供される。本発明に係るデータ保障装置1は、データを保障するための処理を行う。データ保障装置1は、少なくとも、データ取得回路1b、暗号処理回路1cおよびデータ入出力制御回路1eを有する。データ取得回路1bは、処理対象データ2を取得する。暗号処理回路1cは、入力されたデータの暗号処理を行う。データ入出力制御回路1eは、データ取得回路1bに第1のバス1gを介して接続されると共に暗号処理回路1cに対して第2のバス1hを介して接続されており、データ取得回路1bが取得した処理対象データ2を第1のバス1g経由で取得して内蔵メモリに格納し、処理対象データ2を第2のバス1h経由で暗号処理回路1cに入力し、暗号処理回路1cから第2のバス1h経由で暗号処理実行後の処理結果データ3を取得する。

【0029】

このようなデータ保障装置1によれば、処理対象データ2が第1のバス1g経由でデータ入出力制御回路1eで取得され、一旦内蔵メモリに格納される。その後、データ入出力制御回路1eにより処理対象データ2が第2のバス1h経由で暗号処理回路1cに入力される。処理結果データ3は、第2のバス1h経由でデータ入出力制御回路1eが取得する。

【0030】

また、本発明では上記課題を解決するために、保障されたデータをネットワークを介して送受信するデータ通信装置において、送信データを生成するメインCPUと、入力されたデータを暗号化する暗号処理回路と、入力されたデータを前記ネットワークを介して送信する通信回路と、前記メインCPUと前記通信回路に第1のバスを介して接続されると共に前記暗号処理回路に対して第2のバスを介して接続されており、前記メインCPUが取得した前記送信データを前記第1

のバス経由で取得して内蔵メモリに格納し、前記送信データを前記第2のバス経由で前記暗号処理回路に入力し、前記暗号処理回路から前記第2のバス経由で暗号化後の暗号データを取得し、前記通信回路に対して入力するデータ入出力制御回路と、を有することを特徴とするデータ通信装置が提供される。

【0031】

このようなデータ通信装置によれば、メインCPUで生成された送信データが第1のバス経由でデータ入出力制御回路で取得され、データ入出力制御回路により送信データが第2のバス経由で暗号処理回路に入力される。送信データは、第2のバス経由でデータ入出力制御回路が取得し、その送信データは、データ入出力制御回路により第1のバスを介して通信回路に渡される。

【0032】

また、上記課題を解決するために、データを保障するためのデータ保障方法において、データ取得回路で取得した処理対象データを、データ入出力制御回路が第1のバス経由で取得して内蔵メモリに格納し、前記データ入出力制御回路が、前記処理対象データを第2のバス経由で暗号処理回路に入力し、前記暗号処理回路が前記処理対象データの暗号処理を行い、暗号処理実行後の処理結果データを、前記暗号処理回路から前記データ入出力制御回路に渡す、ことを特徴とするデータ保障方法が提供される。

【0033】

このようなデータ保障方法によれば、データ取得回路で取得した処理対象データが、データ入出力制御回路により第1のバス経由で取得される。次に、データ入出力制御回路により、処理対象データが第2のバス経由で暗号処理回路に入力される。そして、暗号処理回路により処理対象データの暗号処理が行われ、暗号処理実行後の処理結果データが、暗号処理回路からデータ入出力制御回路にされる。

【0034】

【発明の実施の形態】

以下、本発明の実施の形態を図面を参照して説明する。

まず、実施の形態に適用される発明の概要について説明し、その後、実施の形

態の具体的な内容を説明する。

【0035】

図1は、実施の形態に適用される発明の概念図である。データ保障装置1は、CPU(Central Processing Unit)1a、データ取得回路1b、暗号処理回路1c、認証処理回路1d、データ入出力制御回路1e、およびデータ出力回路1fを有している。CPU1a、データ取得回路1b、データ入出力制御回路1e、およびデータ出力回路1fは、互いに第1のバス1gで接続されている。暗号処理回路1cとデータ入出力制御回路1eとは、第2のバス1hで接続されている。認証処理回路1dとデータ入出力制御回路1eとは、第3のバス1iで接続されている。

【0036】

CPU1aは、データ保障装置1全体を制御する。データ取得回路1bは、処理対象データ2を取得する。たとえば、ネットワーク経由で入力されるデータを受信する。また、CPU1aは、認証処理回路1dから認証データを受け取った際には、処理対象データ2の認証や処理結果データ3への認証データの付与等の処理を行う。

【0037】

暗号処理回路1cは、入力されたデータの暗号処理を行う。暗号処理とは、データの暗号化処理または復号処理である。たとえば、インターネット等の広域ネットワークを介して送信すべきデータに対しては、暗号化処理が行われる。また、暗号化されたデータを受信した場合には、復号処理が行われる。

【0038】

認証処理回路1dは、データの認証処理を行う。認証処理とは、たとえば、ハッシュ関数に基づくハッシュ値等の認証データの生成処理である。生成された認証データは、たとえば、CPU1aに渡される。

【0039】

データ入出力制御回路1eは、暗号処理回路1cや認証処理回路1dへのデータの入出力を制御する。具体的には、データ入出力制御回路1eは、データ取得回路1bが取得した処理対象データ2を第1のバス1g経由で取得し、内蔵メモ

りに格納する。暗号処理が必要な場合は、データ入出力制御回路 1 e は、処理対象データ 2 を第 2 のバス経由 1 h で暗号処理回路 1 c に入力する。その後、データ入出力制御回路 1 e は、暗号処理回路 1 c から第 2 のバス 1 h 経由で暗号処理実行後の処理結果データ 3 を取得する。また、認証処理が必要な場合は、データ入出力制御回路 1 e は、処理対象データ 2 を第 3 のバス 1 i 経由で暗号処理回路 1 c に入力する。データ入出力制御回路 1 e は、処理結果データ 3 を、データ出力回路 1 f に渡す。

【0040】

なお、データ入出力制御回路 1 e に DMA (Direct Memory Access) コントローラを内蔵させることで、第 1 のバス 1 g を介したデータの受け渡しを DMA によって行うことができる。

【0041】

データ出力回路 1 f は、受け取った処理結果データ 3 を出力する。処理結果データ 3 が暗号化されたデータの場合、その処理結果データ 3 には CPU 1 a 等の制御に従ってハッシュ値等の認証データが付加される。

【0042】

このような構成のデータ保障装置 1 において、以下の様な処理が行われる。なお、データ保障装置 1 で行う処理には、平文のデータを暗号化等により安全性を高めて出力する処理と、暗号データの復号等により内容を参照可能なデータ形式に変換して出力する処理とがある。

【0043】

まず、平文のデータを暗号化して出力する場合の処理を説明する。平文の処理対象データ 2 がデータ取得回路 1 b で受け取られると、その処理対象データ 2 をデータ入出力制御回路 1 e が取得し、内蔵メモリに格納する。そして、データ入出力制御回路 1 e により、内蔵メモリに格納された処理対象データ 2 が暗号処理回路 1 c に入力される。すると、暗号処理回路 1 c により処理対象データ 2 が暗号化されデータ入出力制御回路 1 e に渡される。

【0044】

暗号化された処理結果データ 3 は、データ入出力制御回路 1 e により認証処理

回路 1 d に渡される。すると、認証処理回路 1 d により、認証処理が行われる。たとえば、ハッシュ値等の認証データが生成される。処理結果データ 3 には、データ入出力制御回路 1 e によってデータ出力回路 1 f に渡される。そして、CPU 1 a のデータ処理等により処理結果データ 3 に認証データが付与されデータ出力回路 1 f から出力される。

【0045】

次に、暗号データを復号して出力する場合の処理について説明する。暗号化された処理対象データ 2（認証データが付与されているものとする）がデータ取得回路 1 b で受け取られると、その処理対象データ 2 をデータ入出力制御回路 1 e が取得し、内蔵メモリに格納する。そして、データ入出力制御回路 1 e により、内蔵メモリに格納された処理対象データ 2 が認証処理回路 1 d に入力される。すると、認証処理回路 1 d により、認証処理が行われる。たとえば、ハッシュ値等の認証データが生成される。生成された認証データは、処理対象データ 2 に付与されていた認証データと比較される。比較処理は、たとえば CPU 1 a によって行われる。

【0046】

比較の結果、正しいデータであることが認証されると、データ入出力制御回路 1 e により、内蔵メモリに格納された処理対象データ 2 が暗号処理回路 1 c に入力される。すると、暗号処理回路 1 c により処理対象データ 2 が復号されデータ入出力制御回路 1 e に渡される。平文に復号された処理結果データ 3 は、データ入出力制御回路 1 e によってデータ出力回路 1 f に渡される。そして処理結果データ 3 がデータ出力回路 1 f から出力される。

【0047】

このように、暗号処理や認証処理を行う際、一旦処理対象データをデータ入出力制御回路 1 e の内蔵メモリに格納してしまえば、CPU 1 a が接続された第 1 のバス 1 g とは別の第 2 のバス 1 h や第 3 のバス 1 i を介して、暗号処理回路 1 c や認証処理回路 1 d に対するデータの入出力を行うことができる。その結果、暗号処理回路 1 c や認証処理回路 1 d に対するデータの入出力を、CPU 1 a が管理する必要がなくなり、CPU 1 a の処理負荷が軽減される。

【0048】

しかも、データ入出力制御回路 1 e に DMA コントローラを内蔵することで、第 1 のバス 1 g 経由で行われる処理対象データ 2 の取得や処理結果データ 3 の受け渡しを、データ入出力制御回路 1 e の制御下で行うことができる。たとえば、処理対象データ 2 が大容量の場合、処理対象データ 2 を予め他のメモリに格納し、データ入出力制御回路 1 e はそのメモリから DMA 転送により処理対象データ 2 を取得することができる。また、データ入出力制御回路 1 e は、処理結果データ 3 についても、DMA 転送により他のメモリに転送することができる。その結果、CPU 1 a の処理負荷が更に軽減される。

【0049】

なお、データ取得回路 1 b とデータ出力回路 1 f との機能を併せ持つ通信インタフェースをデータ保障装置 1 に含めることで、データの保障機能を有するデータ通信装置が構成される。このようなデータ通信装置は VPN 装置と同等の機能を実現している。以下、VPN 機能を有するデータ保障装置 1 をセキュリティネットワークコントローラと呼ぶこととする。

【0050】

セキュリティネットワークコントローラを各種電子機器に実装すれば、その電子機器は、安全なデータ通信を容易に行うことができる。たとえば、本発明を適用したセキュリティネットワークコントローラをカメラシステム（デジタル画像の撮像機能と撮像された画像データを処理する機能とを有するコンピュータシステム）に実装することで、インターネットを介したオンラインの動画配信を、所定の利用者（たとえば、予め登録された会員）に対して行うことができる。また、遠隔地からのカメラの制御を、安全に行うことも可能となる。

【0051】

なお、IPsec の機能を利用すれば、予め決められた利用者以外の一般の利用者との間でセキュアな通信を行うことができる。すなわち、IPsec では、ISAKMP (Internet Security Association Key Management Protocol) を利用することで、安全に相手の認証や鍵の交換ができる。そのため、最初に ISAKMP による鍵交換を行えば、任意の装置間でセキュアな通信が可能となる。

【0052】

以下、本発明を適用したセキュリティネットワークコントローラをカメラに実装した場合を例に採り、本発明の実施の形態について詳細に説明する。なお、以下の実施の形態では暗号アルゴリズムにDES (Data Encryption Standard)、ハッシュ関数アルゴリズムにHMAC-SHA1を用いるものとする。

【0053】**[第1の実施の形態]**

図2は、第1の実施の形態に係るシステム構成例を示す図である。図2に示すように、セキュリティネットワークコントローラを実装したカメラシステム10がインターネット20を介して端末装置30に接続されている。端末装置30は、たとえばVPN機能を有するコンピュータである。ここで、端末装置30にもセキュリティネットワークコントローラを実装することで、インターネット20を介した暗号通信が容易となる。

【0054】

図3は、カメラ内部の回路構成を示す図である。カメラシステム10には、セキュリティネットワークコントローラ100、メインCPU11、周辺の回路12、13およびカメラ機構部14が設けられている。セキュリティネットワークコントローラ100は、VPN機能を有しており、ハードウェア的に暗号化／復号等の処理を行うことができる。

【0055】

なお、セキュリティネットワークコントローラ100は、LSIチップ上に構成することができる。LSIチップでセキュリティネットワークコントローラ100を実現することで、各種電子機器への組み込みが容易となる。

【0056】

また、セキュリティネットワークコントローラ100は、1つのモジュール（たとえば、PCMCIA (Personal Computer Memory Card International Association)等のカードモジュール）内に構成することができる。これにより、所定の通信インタフェースを介してカメラシステム10や他の機器に容易に実装することができる。

【0057】

メインCPU11は、周辺の回路12, 13やカメラ機構部14を制御することでカメラシステム10全体を制御している。回路12, 13は、カメラの焦点制御回路や、画像データの圧縮回路等である。また、メインCPU11は、カメラシステム10で撮影したデジタルの画像をセキュリティネットワークコントローラ100に転送する。

【0058】

セキュリティネットワークコントローラ100は、CPU101、メモリコントローラ102、メモリ103、外部接続インタフェース104、通信インタフェース105、IPsec制御回路110、暗号処理回路120、およびハッシュ関数処理回路130を有している。これらの各要素は、内部のバス等で互いに接続されている。

【0059】

CPU101は、セキュリティネットワークコントローラ100全体を制御する。

メモリコントローラ102は、メモリ103へのデータの入出力を制御する。メモリ103は、データを記憶する。なお、図3では、メモリ103をセキュリティネットワークコントローラ100内部に示しているが、外部に接続することもできる。その場合、メモリコントローラ102に外部接続用のコネクタが設けられ、そのコネクタにメモリ103を接続することで、メモリ103がセキュリティネットワークコントローラ100に繋がる。メモリ103としては、SRAM(Static Random Access Memory)やフラッシュメモリ等の半導体メモリを使用することができる。メモリコントローラ102は、メモリ103に対して受信データを書き込んだり、メモリ103内のデータを読み出したりする。メモリ103は、処理対象データの蓄積や、演算時のワークメモリとしても使用される。

【0060】

外部接続インタフェース104は、メインCPU11に接続されており、メインCPU11とCPU101の間の通信データを中継する。外部接続インタフェース104は、基本的にはレジスタとデータで構成され、双方向のデータの受け

渡しを行う。

【0061】

通信インタフェース105は、インターネット20に接続されており、インターネット20を介した通信データを中継する。ここで、通信インタフェース105は、MAC(Media Access Control)とも呼ばれ、イーサネット(登録商標)などの物理層を介したネットワーク接続を実現させる機能を有する。

【0062】

IPsec制御回路110は、暗号処理回路120とハッシュ関数処理回路130へのデータの入出力を制御する。暗号処理回路120は、データの暗号化または復号処理を行う。ハッシュ関数処理回路130は、通信されるデータに基づいてハッシュ値を生成する。

【0063】

このようなシステム構成において、メインCPU11が、制御対象となるカメラを制御する。その際、メインCPU11は、インターネット20を介してセキュアに通信を行うことで、遠隔地からの指示に従った制御を行ったり、カメラで撮影した画像をインターネット20を介して配信したりする。セキュアな通信には、IPsecが用いられる。なお、セキュリティネットワークコントローラ100は、カメラに限らず、電化製品、家電、AV機器等を実装して、それらを制御することができる。

【0064】

通信インタフェース105あるいはIPsec制御回路110は、IPsec処理におけるCPU101の負荷を軽減する機能を有する。その機能により、パケット単位で暗号及び認証処理がCPU101の代わりに実現される。

【0065】

たとえば、データを送信する際には、以下の処理が行われる。

メインCPU11からインターネット20上へ配信すべきデータは、外部接続インタフェース104を介して、メモリ103に転送される。そして、暗号化対象データのDMA転送の指示がCPU101からIPsec制御回路110に出され、IPsec制御回路110が暗号化対象データをDMA転送で取得し、暗

号処理回路120に渡す。暗号処理回路120が、IPsecプロトコルに従って暗号化処理を行う。暗号化されたデータは、IPsec制御回路110によりメモリ103へDMA転送される。続けて、認証用データのDMA転送の指示がCPU101からIPsec制御回路110に出される。すると、IPsec制御回路110が認証用データをDMA転送で取得し、ハッシュ関数処理回路130に渡す。ハッシュ関数処理回路130がIPsecプロトコルに従ってハッシュ値を生成する。そして、CPU101により、暗号化されたデータと、生成されたハッシュ値とを含むパケットが生成され、通信インタフェース105を介してインターネット20上へ送信される。

【0066】

このような手順で画像を配信すれば、所定の端末装置（たとえば、会員登録をしたユーザの端末装置）でのみ再生可能な映像ストリームを配信できる。

また、メインCPU11への制御データがインターネットを介してIPsec処理されたパケットによって送られてくると、以下の処理が行われる。

【0067】

送られてきたパケットを通信インタフェース105が受信する。そのパケットは、CPU101の制御によりメモリ103に書き込まれる。そして、パケット内の認証データのDMA転送の指示がCPU101からIPsec制御回路110に出される。すると、IPsec制御回路110が認証データをDMA転送で取得し、ハッシュ関数処理回路130に渡す。ハッシュ関数処理回路130がIPsecプロトコルに従ってハッシュ値を生成する。CPU101は、ハッシュ値に基づいて、受信したデータを認証する。

【0068】

生成されたハッシュ値により正しく認証されれば、パケット内の暗号データのDMA転送の指示がCPU101からIPsec制御回路110に出される。すると、IPsec制御回路110が暗号データをDMA転送で取得し、暗号処理回路120に渡す。暗号処理回路120が、IPsecプロトコルに従って復号処理を行う。復号された平文のデータは、IPsec制御回路110によりメモリ103へDMA転送される。その後、復号されたデータは、外部接続インタフ

ェース104を介してメインCPU11に送られ、メインCPU11で処理される。

【0069】

このような手順で、カメラの制御指示等をメインCPU11に渡せば、遠隔地からのカメラ制御を安全に行うことができる。

以下に、データの送信および受信の手順を、フローチャートを参照して説明する。

【0070】

図4は、データ送信の手順を示すフローチャートである。以下、図4に示す処理をステップ番号に沿って説明する。

【ステップS11】メインCPU11が映像取得等のアプリケーションを実行し、データを生成する。セキュリティネットワークコントローラ100に渡す。

【0071】

【ステップS12】CPU101の制御により、データをメモリ103に格納する。

【ステップS13】CPU101がカプセル化を行う。カプセル化とは、データをヘッダやトレーラで包み込むことである。

【0072】

【ステップS14】IPsec制御回路110によりデータが暗号処理回路120に渡される。そして、暗号処理回路120がデータの暗号化を行う。暗号化されたデータは、IPsec制御回路110によりメモリ103に戻される。

【0073】

【ステップS15】IPsec制御回路110により暗号化されたデータがハッシュ関数処理回路130に渡される。そして、ハッシュ関数処理回路130がハッシュ値を生成する。生成したハッシュ値は、CPU101に渡される。

【0074】

【ステップS16】CPU101が送信用のフレームを生成する。

【ステップS17】CPU101は、フレームを出力バッファ（通信インタフェース105内のバッファ）に書き込む。

【0075】

〔ステップS18〕通信インタフェース105がフレームをインターネット20上へ送信する。

以下に、動画データを配信する際のデータの流れを、図5～図12を参照して説明する。

【0076】

図5は、データ配信の第1のステップを示す図である。まず、カメラシステム10が撮影した画像のデータ41は、メインCPU11からセキュリティネットワークコントローラ100に入力される。データ41は、外部接続インタフェース104で受け取られ、メモリコントローラ102に転送される。そして、データ41は、メモリコントローラ102によってメモリ103に書き込まれる。

【0077】

図6は、データ配信の第2のステップを示す図である。IPsec制御回路110は、DMA転送により、メモリ103からデータ41を取得する。

図7は、データ配信の第3のステップを示す図である。IPsec制御回路110は、取得したデータ41を暗号処理回路120に渡す。暗号処理回路120は、データ41を暗号化する。そして、暗号処理回路120は、暗号化されたデータ42を、IPsec制御回路110に渡す。この間、メインのバスが解放されているため、後続のデータ43をメインCPU11から受け取り、メモリ103に転送することができる。

【0078】

図8は、データ配信の第4のステップを示す図である。IPsec制御回路110は、暗号化されたデータ42を、DMA転送により、メモリコントローラ102に渡す。メモリコントローラ102は、受け取ったデータ42をメモリ103に格納する。

【0079】

図9は、データ配信の第5のステップを示す図である。IPsec制御回路110は、DMA転送により、メモリ103から暗号化されたデータ42を取得する。

【0080】

図10は、データ配信の第6のステップを示す図である。IPsec制御回路110は、暗号化されたデータ42をハッシュ関数処理回路130に渡す。ハッシュ関数処理回路130は、データ42に対してハッシュ関数を適用し、ハッシュ値を生成する。この間、メインのバスが解放されているため、後続のデータ44をメインCPU11から受け取り、メモリ103に転送することができる。

【0081】

図11は、データ配信の第7のステップを示す図である。ハッシュ関数処理回路130は、生成したハッシュ値45をCPU101に渡す。CPU101は、メモリ103に格納されている暗号化後のデータ42にハッシュ値45を付加する。

【0082】

図12は、データ配信の第8のステップを示す図である。CPU101は、メモリ103に格納されている暗号化後のデータ42とハッシュ値45とから通信用のパケット46を生成し、通信インタフェース105を介して、インターネット20に接続された端末装置30へ配信する。

【0083】

このように、メモリ103とIPsec制御回路110との間のデータの受け渡しは、IPsec制御回路110が有するDMA機能によって行われるため、CPU101の負荷が少なくて済む。また、IPsec制御回路110と暗号処理回路120との間が専用のバスで接続され、暗号化処理のためのデータの受け渡しが専用のバスを介して行われるため、その間メインのバスを他のデータ転送に利用できる。同様に、IPsec制御回路110とハッシュ関数処理回路130との間が専用のバスで接続され、ハッシュ処理のためのデータの受け渡しが専用のバスを介して行われるため、その間メインのバスを他のデータ転送に利用できる。その結果、処理効率が向上する。

【0084】

図13は、データの受信手順を示すフローチャートである。以下、図13に示す処理をステップ番号に沿って説明する。

〔ステップS21〕通信インタフェース105が、インターネット20を経由して送られたフレームを受信する。

【0085】

〔ステップS22〕受信したフレームに含まれるパケットが通信インタフェース105内部の入力バッファに格納される。

〔ステップS23〕CPU101により、パケットのヘッダ処理が行われる。

【0086】

〔ステップS24〕IPsec制御回路110によりデータが暗号処理回路120に渡される。そして、ハッシュ関数処理回路130がハッシュ値を生成する。認証処理が行われる。CPU101は、ハッシュ関数処理回路130で生成されたハッシュ値と受信したデータに付加されていたハッシュ値とを比較し、データの認証を行う。

【0087】

〔ステップS25〕認証されたら、IPsec制御回路110によりデータが暗号処理回路120に渡される。そして、暗号処理回路120が復号処理を行う。

【0088】

〔ステップS26〕CPU101は、カプセル化されたデータを解析する（ヘッダやトレーラを取り除く）。

〔ステップS27〕CPU101は、データをメモリ103に格納する。

【0089】

〔ステップS28〕CPU101は、データを外部接続インタフェース104を介してメインCPU11に渡す。メインCPU11は、アプリケーションプログラムに従ってデータを処理する。

【0090】

なお、データ受信の場合のデータの流れについては、図5～図12に示したデータ送信の場合の逆の手順である。

このように、データを受信した場合でも、暗号処理回路120およびハッシュ関数処理回路130へのデータの入出力は、IPsec制御回路110が行う。

また、IPsec制御回路110と他の回路との間のデータの受け渡しは、DMA転送によって行われる。

【0091】

次に、IPsec制御回路110とその周辺の回路との接続関係および機能について詳細に説明する。

図14は、セキュリティネットワークコントローラの内部構成例を示す図である。セキュリティネットワークコントローラ100は、CPU101、バスセクタ107、暗号処理回路120、データインセクタ106、IPsec制御回路110、ハッシュ関数処理回路130、メモリコントローラ102、およびメモリ103を有している。

【0092】

CPU101は、バス181を介してバスセクタ107に接続されている。CPU101は、バスセクタ107を経由して他の構成要素との間で情報の交換を行い、セキュリティネットワークコントローラ100全体を制御する。また、CPU101には、暗号処理回路120、IPsec制御回路110、およびハッシュ関数処理回路130から外部割込み信号190が入力される。さらに、CPU101には、IPsec制御回路110から外部バス開放要求受付信号191が入力される。CPU101からIPsec制御回路110とバスセクタ107には、外部バス開放要求信号192が入力される。

【0093】

バスセクタ107は、バス182を介して、暗号処理回路120、IPsec制御回路110、ハッシュ関数処理回路130、およびメモリコントローラ102に接続されている。さらに、バスセクタ107は、専用のバス183でIPsec制御回路110に接続されている。バスセクタ107は、CPU101からの制御に従って、バス182に接続された各要素に対してCPU101からのデータを送信したり、各要素から送られたデータをCPU101へ転送したりする。

【0094】

具体的には、バスセクタ107は、CPU101とIPsec制御回路11

0 とのうちバスマスタになっている方から出力される制御信号等を選択し、他の回路に対して出力する。出力される制御信号等は、たとえば、他の回路に対するアドレス、制御信号、ライトデータである。

【0095】

暗号処理回路 120 は、データの暗号化および復号を行う回路である。第 1 の実施の形態では、DES による暗号化／復号を行う。暗号処理回路 120 は、専用のバス 184 で、IPsec 制御回路 110 に接続されている。また、暗号処理回路 120 は、専用のバス 185 で、データインセクタ 106 に接続されている。暗号処理回路 120 は、バス 184 を介して IPsec 制御回路 110 から暗号化または復号対象のデータを取得する。そして、取得したデータの暗号化または復号を行い IPsec 制御回路 110 に渡す。

【0096】

なお、第 1 の実施の形態では、暗号処理回路 120 は、64 ビットブロック暗号処理を行う。暗号処理回路 120 は、64 ビットデータがライトされると内部処理ステートマシンのスタートビットが、自動的にオンされる。これにより、処理対処のデータの書き込み後、すぐに暗号化または復号処理を開始することができる。

【0097】

また、暗号処理回路 120 の行う暗号化または復号の処理は、16 ラウンド処理である。そのため、内部カウンタにより 16 が数えられると内部ステートマシンの終了信号がオンとなる。この終了信号が IPsec 制御回路 110 に出力される。

【0098】

なお、暗号処理回路 120 は、IPsec 制御回路 110 を介さずにデータの暗号化や復号処理を行うことができる。その場合、暗号処理回路 120 は、終了信号のオンに替えて、外部割込み信号 190 を CPU 101 へ出力する。

【0099】

データインセクタ 106 は、既に説明した接続関係以外に、IPsec 制御回路 110 に接続されている。また、データインセクタ 106 は、バス 187

を介してハッシュ関数処理回路130に接続されている。さらに、データインセクタ106は、バス188を介してCPU101、メモリコントローラ102、およびIPsec制御回路110に接続されている。データインセクタ106は、暗号処理回路120、IPsec制御回路110およびハッシュ関数処理回路130からのバス185, 186, 187を介して入力される信号の1つを選択して、バス188を介してCPU101、メモリコントローラ102、およびIPsec制御回路110に対して出力する。

【0100】

IPsec制御回路110は、前述のようにバス184を介して暗号処理回路120に接続されており、更に、バス189を介してハッシュ関数処理回路130に接続されている。IPsec制御回路110は、256バイトの内蔵RAMを有しており、暗号化／復号対象のデータや、ハッシュ処理対象のデータを保持することができる。

【0101】

また、IPsec制御回路110は、DMAC（ダイレクトメモリアクセスコントローラ）によるバスアービター機能を内蔵しており、CPU101とのバス・アービトレーションを行い、バスマスタとなることができる。バスマスタとなることで、IPsec制御回路110は、DMA転送用の受信先アドレス、送信先アドレス、転送量、モードの設定を行うことができる。

【0102】

IPsec制御回路110は、基本的に、バスマスタになると処理前のデータを受信先より取り込み、処理後のデータを送信先に書き込む動作を行う。そして、IPsec制御回路110は、データ転送完了後CPU101にバス権を返還する。このように、IPsec制御回路110は、DMACの機能を用いて、暗号化／復号の対象となるデータまたはハッシュ関数による認証処理の対象となるデータを、DMA機能を用いてメモリ103から取得する。

【0103】

暗号化／復号の処理対象となるデータを取得した場合、IPsec制御回路110は、そのデータを暗号処理回路120に渡す。その後、IPsec制御回路

110は、暗号化または復号処理が行われたデータを受け取り、DMA処理によりメモリ103に転送する。また、認証の対象となるデータを取得した場合、IPsec制御回路110は、そのデータをハッシュ関数処理回路130に渡す。なお、IPsec制御回路110は、データの取り込み及び書き出し処理を、暗号処理回路120による暗号化／復号処理や、ハッシュ関数処理回路130による認証処理と並行化して実行する。

【0104】

また、IPsec制御回路110は、内部に処理対象データ長を格納するレジスタを持ち、処理ごとに処理終了分の長さを減算し、処理対象データの残り分を管理する。そして、IPsec制御回路110は、全ての処理が終わった際に処理終了割り込み信号をCPU101へ出力する。

【0105】

ハッシュ関数処理回路130は、ハッシュ関数を用いてハッシュ値を生成する回路である。具体的には、ハッシュ関数処理回路130は、IPsec制御回路110から受け取ったデータに基づいてハッシュ値を生成し、CPU101に渡す。

【0106】

なお、第1の実施の形態では、ハッシュ関数処理回路130は、512ビットブロック処理を行う。そのため、512ビットのデータがライトされると、内部処理ステートマシンのスタートビットがオンされる。これにより、ハッシュ値の生成処理が開始される。

【0107】

また、ハッシュ関数処理回路130は、80ラウンドの処理を行う。そこで、ハッシュ関数処理回路130は、内部カウンタにより80が数えられると内部ステートマシンの終了信号をオンにする。オンにされた終了信号は、IPsec制御回路110に出力される。

【0108】

なお、ハッシュ関数処理回路130は、IPsec制御回路110を介さずにデータの暗号化や復号処理を行うことができる。その場合、終了信号に替えて、

外部割込み信号 190 が CPU 101 へ出力される。

【0109】

メモリコントローラ 102 は、メモリ 103 に対するデータの書き込みやデータの読み出しを行う。

メモリ 103 は、メモリコントローラ 102 に接続されている。メモリ 103 は、SRAM やフラッシュメモリなどの読み出しと書き込みとが可能な半導体の記録媒体である。

【0110】

このような構成において、IPsec 制御回路 110 と暗号処理回路 120 とを組み合わせる複合的に動作させる時には、以下のような処理が行われる。

まず、IPsec 制御回路 110 が外部バス開放要求信号 192 を CPU 101 へ出力する。その後、CPU 101 が外部バス開放要求受付信号 191 を出力し、バス権が IPsec 制御回路 110 へ開放される。これにより、CPU 101 を介在させない、暗号化または復号処理が開始される。

【0111】

IPsec 制御回路 110 は、メモリ 103 内から DMA 転送によってデータを取得する。取得したデータは、IPsec 制御回路 110 内のメモリに格納される。なお、IPsec 制御回路 110 内のメモリは、2 面構成となっている。1 つの面には暗号化または復号処理前のデータが格納され、他方の面には、暗号化または復号処理後のデータが格納される。

【0112】

第 1 の実施の形態では、暗号処理回路 120 において DES 処理が行われる。DES 処理は、64 ビット単位で実行される。したがって、IPsec 制御回路 110 がメモリ 103 からデータを取得する場合、64 ビットの整数倍のデータ長の DMA 転送を行う。第 1 の実施の形態では、64 バイト（DES 処理 8 回分）のデータを一度に取得するものとする。

【0113】

図 15 は、IPsec 制御回路の内蔵 RAM の DES 処理時の構成を示す図である。図 15 に示すように、IPsec 制御回路 110 の内蔵 RAM 111 には

、A面111aとB面111bとがある。A面111aとB面111bとには、それぞれDES処理16回分の処理対象データ(128バイト)が格納できる。A面111aは演算対象データの格納領域であり、B面111bは演算結果データの格納領域である。

【0114】

メモリ103からDMA転送によって取得されたDES処理前のデータは、A面111aに格納される。たとえば、64バイトのデータがDMA転送でA面111aの領域「A0」に格納される。領域「A0」に格納された64バイトのデータが先に暗号処理回路120に渡されDES処理が行われる。

【0115】

領域「A0」に格納されたデータに対するDES処理が行われている間に、領域「A1」に後続の64バイトのデータがDMA転送で格納される。DES処理が施されたデータが暗号処理回路120から出力されると、B面111bの64バイトの領域「B0」に格納される。

【0116】

続けて、A面111aの64バイトの領域「A1」に格納されたデータが暗号処理回路120に渡されDES処理が行われる。DES処理が施されたデータが暗号処理回路120から出力されると、B面111bの64バイトの領域「B1」に格納される。

【0117】

B面111bの「B0」と「B1」との一方の領域に処理済みのデータが格納されると、B面111b内のデータがIPsec制御回路110によりメモリ103にDMA転送される。

【0118】

IPsec制御回路110がデータを受信する場合、IPsec制御回路110から外部バス開放要求信号192が出力される。IPsec制御回路110は、データの取り込みが終了したらバス権をCPU101へ返還する。それと同時に暗号処理回路120とIPsec制御回路110が専用バスと専用制御信号を用いてCPU101を介在せずに処理を開始する。この際、内蔵RAM111の

アドレスはインクリメント構造を取っている。暗号処理されたデータは I P s e c 制御回路 110 の 128 バイトの内蔵 R A M 111 の B 面 111 b に貯められる。この際、格納先を指し示す内蔵 R A M 111 のアドレスは、インクリメント構造を取っている。すなわち、前回格納時のアドレスに所定の値（64 バイト分）を加算したアドレスが指定される。

【0119】

64 バイト処理が終了するごとに、I P s e c 制御回路 110 は外部バス開放要求信号 192 を C P U 101 へ出力する。I P s e c 制御回路 110 は、バス権獲得後にメモリ 103 へ演算処理結果を書き出す。書き出しが終了するとバス権を C P U 101 へ返還する。

【0120】

なお、暗号処理回路 120 による演算中も演算対象データを格納している内蔵 R A M 111 の中の 64 バイト処理が終了するごとに外部バス開放要求信号 192 が出力される。バス権が I P s e c 制御回路 110 へ開放されてからメモリ 103 から、暗号対象データを取り込む。暗号化対象データの取り込みが終了するとバス権が C P U 101 へ返還される。

【0121】

また、暗号処理回路 120 による演算中も演算終了データを格納するための B 面 111 b の中に 64 バイトが格納されるごとに、外部バス開放要求信号 192 が出力される。バス権が I P s e c 制御回路 110 へ開放されてから、メモリ 103 へ演算処理結果が書き出される。書き出しが終了するとバス権を C P U 101 へ返還する。

【0122】

一方、I P s e c 制御回路 110 とハッシュ関数処理回路 130 とを組み合わせる複合的に処理を行う際には、内蔵 R A M 111 を 128 バイトの 2 面構成として使用する。この場合、2 面とも全て書き込み用として使用される。演算終了データとして得られるハッシュ値は、ハッシュ関数処理回路 130 内部の 160 ビットのハッシュ値格納レジスタに格納される。そして、演算終了後に、ハッシュ値格納レジスタから C P U 101 にハッシュ値が渡される。なお、第 1 の実施

の形態では、ハッシュ処理として、SHA1処理を行う。

【0123】

図16は、IPsec制御回路の内蔵RAMのSHA1処理時の構成を示す図である。図16に示すように、内蔵RAM111のA面111aとB面111bとが、それぞれ128バイトの1つの領域として扱われる。

【0124】

IPsec制御回路110は、内蔵RAM111にメモリ103から、ハッシュ関数による処理対象データをDMA転送により取り込む。この際128バイト連続で書き込んでからバス権をCPU103に返還する。その後、CPU103を介在させず、IPsec制御回路110とハッシュ関数処理回路130とでハッシュ処理を開始する。

【0125】

IPsec制御回路110は、演算中、片面128バイトが空であれば、すぐにメモリ103から、ハッシュ関数による処理対象データを取り込む。この際128バイト連続で書き込んでからバス権を返還する。なお、内蔵RAM111のアドレスはインクリメント構造を取っている。

【0126】

IPsec制御回路110、暗号処理回路120、ハッシュ関数処理回路130などから出力されるリードデータは、データインセクタ106を通過してCPU101、IPsec制御回路110に出力されるか、メモリコントローラ102を通過してメモリ103に出力される。

【0127】

次に、IPsec制御回路110と暗号処理回路120、ハッシュ関数処理回路130の動作について詳しく説明する。

図17は、IPsec制御回路の内部構成を示す図である。図17に示すように、IPsec制御回路110には、内蔵RAM111、内蔵RAMインタフェース112、レジスタ群113、スレーブバスインタフェース114、マスタバスインタフェース115、およびマクロインタフェース116を有している。

【0128】

内蔵RAM 111は、図13、図14に示した通りであり、処理対象データや処理結果データが格納される。内蔵RAMインタフェース112は、内蔵RAM 111へのデータの格納や読み出しを行うインタフェースである。レジスタ群113は、制御データ等を格納するための複数の汎用レジスタである。スレーブバスインタフェース114は、バス182上でスレーブとして動作するためのインタフェースである。マスタバスインタフェース115は、バス182上でバスマスタとして動作するためのインタフェースである。マスタバスインタフェース115により、バス182を介したDMA転送が行われる。マクロインタフェース116は、暗号処理回路120やハッシュ関数処理回路130と通信するためのインタフェースである。

【0129】

IPsec制御回路110と暗号処理回路120とは、ライト要求信号(DREQ__WR)、リード要求信号(DREQ__RD)、ライトストロープ信号(IP__WRX[0])、リードストロープ信号(IP__RDX)、アドレス信号(IP__A)、リードデータバス(IP__RD[31:0])、ライトデータバス(IP__WD[31:0])で接続されている。また、IPsec制御回路110とハッシュ関数処理回路130とは、ライトデータバス(IP__WD[31:0])、ライトストロープ信号(IP__WRX[1])、ライト要求信号(DREQ__WR)で接続されている。

【0130】

ライト要求信号(DREQ__WR)は、暗号処理回路120からIPsec制御回路110へ32ビットのデータライトを要求する信号である。

リード要求信号(DREQ__RD)は、暗号処理回路120からIPsec制御回路110へ32ビットのデータリードを要求する信号である。

【0131】

ライトストロープ信号(IP__WRX[0])は、IPsec制御回路110から暗号処理回路120へデータのライトを通知する信号である。

リードストロープ信号(IP__RDX)は、IPsec制御回路110から暗号処理回路120へ、暗号処理回路120内のデータのリードを通知するための信号である。

【0132】

アドレス信号(IP_A)は、IPsec制御回路110から暗号処理回路120へデータのアクセス先のアドレスを指定するための信号である。

リードデータバス(IP_RD[31:0])は、32ビットの暗号処理回路120からのリードデータをIPsec制御回路110に渡す専用バスである。

【0133】

ライトデータバス(IP_WD[31:0])は、IPsec制御回路110から暗号処理回路120またはハッシュ関数処理回路130へ、32ビットのライトデータを渡す専用バスである。

【0134】

ライトストロブ信号(IP_WRX[1])は、IPsec制御回路110からハッシュ関数処理回路130へ、データのライトを通知するための信号である。

ライト要求信号(DREQ_WR)は、ハッシュ関数処理回路130からIPsec制御回路110へ、データライトを行うように要求するための信号である。

【0135】

図18は、暗号処理回路の内部構成を示す図である。暗号処理回路120には、鍵レジスタ121、16進カウンタ122、演算部123、64ビットデータレジスタ群124、2進カウンタ125が設けられている。

【0136】

暗号処理回路120にはクロック信号、アドレス信号(IP_A)、ライトデータバス(IP_WD[31:0])、ライトストロブ信号(IP_WRX[0])が入力されている。暗号処理回路120は、クロック信号に同期して動作する。アドレス信号(IP_A)は、鍵レジスタ121、演算部123、および64ビットデータレジスタ群124に入力される。ライトデータバス(IP_WD[31:0])は、鍵レジスタ121と64ビットデータレジスタ群124に入力される。ライトストロブ信号(IP_WRX[0])は、鍵レジスタ121、演算部123、64ビットデータレジスタ群124および2進カウンタ125に入力される。

【0137】

鍵レジスタ121は、暗号化または復号の際に使用する鍵データ（暗号鍵また

は復号鍵)を格納するためのレジスタである。鍵レジスタ121に格納されたデータは、演算部123に入力される。

【0138】

16進カウンタ122は、演算部123から入力される演算イネーブル信号に基づいて、演算回数をカウントする。演算回数が16未満の間、演算中を示す演算ステータスが演算部123に入力される。また、16進カウンタ122で16までカウントすると、演算終了信号が出力される。

【0139】

演算部123は、暗号化または復号の演算を行う。具体的には、演算部123は、2進カウンタ125からの演算開始信号の入力を受けて、演算を開始する。演算部123は、演算を行う場合、まず、演算データを64ビットデータレジスタ群124から取得する。そして、演算部123は、鍵レジスタ121から入力される鍵データを用いて、演算データの暗号化または復号を行う。演算終了後、演算部123は演算結果を64ビットデータレジスタ群124内のレジスタに格納する。なお、演算結果は、演算データが格納されていたレジスタへ上書きで格納される。

【0140】

64ビットデータレジスタ群124は、演算データや演算結果を格納するためのレジスタ群である。具体的には、64ビットデータレジスタ群124は、2つの32ビットレジスタで構成される。64ビットデータレジスタ群124には、32ビットのライトデータバスを介して演算データが入力され、一方のレジスタに格納される。また、64ビットデータレジスタ群124に演算結果が格納されるとリードデータバス(IP_RD[31:0])として出力される。

【0141】

2進カウンタ125は、ライトストローク信号の入力回数をカウントするカウンタである。2進カウンタ125は、ライトストローク信号が2回入力されると、演算開始信号を演算部123に対して出力する。すなわち、ライトデータバスのバス幅が32ビットであるため、2回の書き込みで64ビットデータレジスタ群124への演算データの格納が完了する。そこで、ライトストローク信号が2

回入力されることで、64ビット単位のDES処理が開始可能となる。

【0142】

このような暗号処理回路120でDES処理を行う場合、まず、鍵レジスタ121に鍵データが格納される。次に、ライトデータバスを介して、64ビットデータレジスタ群124に演算データが2回に分けて書き込まれる。2回の書き込みが行われたことが2進カウンタ125で検出され、演算開始信号が出力される。

【0143】

演算開始信号に応じて、演算部123が演算データに対するDES処理を行い、演算結果を64ビットデータレジスタ群124に書き戻す。演算結果は、64ビットデータレジスタ群124からリードデータとして出力される。また、演算部123によってDES処理が行われる毎に、16進カウンタ122に対してイネーブル信号が出力され、16進カウンタの値がカウントアップされる。そして、16進カウンタ122の値が16に達すると、演算終了信号が出力される。

【0144】

このように、第1の実施の形態に係る暗号処理回路120では、データライト数(32ビット単位)で数えており、データライト終了と共に演算開始を自動認識し、演算を開始することができる。すなわち、2進カウンタ125がデータライトカウンタとして機能しており、このデータライトカウンタが処理開始を検知する。また、16進カウンタ122が演算ラウンドカウンタとして機能しており、所定回数の演算ラウンドの終了により演算終了を認識し、演算終了信号を出力する。なお、暗号化または復号の演算結果は、64ビットデータレジスタ群124(32ビット×2)に上書きされる。

【0145】

次に、ハッシュ関数処理回路130の内部構成について説明する。

図19は、ハッシュ関数処理回路の内部構成を示す図である。ハッシュ関数処理回路130は、ハッシュ値格納レジスタ131、80進カウンタ132、演算部133、512ビットデータレジスタ群134、および16進カウンタ135を有している。

【0146】

ハッシュ関数処理回路130には、クロック信号、アドレス信号、ライトデータ、およびライトストロブ信号が入力されている。ハッシュ関数処理回路130は、クロック信号に同期して動作する。アドレス信号は、演算部133と512ビットデータレジスタ群134とに入力される。ライトデータは、演算部133と512ビットデータレジスタ群134とに入力される。ライトストロブ信号は、演算部133、512ビットデータレジスタ群134および16進カウンタ135に入力される。

【0147】

ハッシュ値格納レジスタ131は、ハッシュ関数処理により生成されたハッシュ値を格納するレジスタ群である。第1の実施の形態では、5個の32ビットレジスタで構成されており、全体で160ビットのデータが格納できる。

【0148】

80進カウンタ132は、演算部133から入力される演算イネーブル信号に基づいて、演算回数をカウントする。演算回数が80未満の間、演算中を示す演算ステータスが演算部133に入力される。また、80進カウンタ132で80までカウントすると、演算終了信号が出力される。

【0149】

演算部133は、ハッシュ関数に基づくハッシュ値の算出処理を行う。具体的には、演算部133は、16進カウンタ135からの演算開始信号の入力を受けて、演算を開始する。演算部133は、演算を行う場合、まず、演算データを512ビットデータレジスタ群134から取得する。そして、演算部133は、演算データに基づいてハッシュ値を生成する。演算終了後、演算部133は演算結果を512ビットデータレジスタ群124内のレジスタに格納する。なお、演算結果は、演算データが格納されていたレジスタへ上書きで格納される。

【0150】

512ビットデータレジスタ群134は、演算データや演算結果を格納するためのレジスタ群である。具体的には512ビットデータレジスタ群134は、16個の32ビットレジスタで構成される。512ビットデータレジスタ群134

には、32ビットのライトデータバスを介して演算データが入力され、順次各レジスタに格納される。

【0151】

16進カウンタ135は、ライトストローク信号の入力回数をカウントするカウンタである。16進カウンタ135は、ライトストローク信号が16回入力されると、演算開始信号を演算部133に対して出力する。すなわち、ライトデータバスのバス幅が32ビットであるため、16回の書き込みで512ビットデータレジスタ群134への演算データの格納が完了する。そこで、ライトストローク信号が16回入力されることで、512ビット単位のハッシュ関数処理が開始可能となる。

【0152】

このようなハッシュ関数処理回路130において、ハッシュ関数に基づいてハッシュ値を生成する場合、まず、512ビットデータレジスタ群134に、処理対象のデータが32ビットずつ格納される。書き込み回数は、16進カウンタ135によってカウントされており、データの書き込みが16回に達すると16進カウンタ135から演算開始信号が出力される。演算開始信号に応答して、演算部133がハッシュ関数に従った演算を開始する。具体的には、演算部133は、512ビットデータレジスタ群134から処理対象のデータを取得し、ハッシュ関数に従った処理を行う。演算部133による処理が80回繰り返されることでハッシュ値が生成され、ハッシュ値格納レジスタ131にハッシュ値が格納される。ハッシュ値格納レジスタ131内のデータは、リードデータとして出力される。このとき、演算が80回に達したことを80進カウンタ132が検出し、演算終了信号を出力する。

【0153】

以上のようにして、ハッシュ関数処理回路130において、ハッシュ値が生成される。

ここで、CPU101、IPsec制御回路110、暗号処理回路120、およびハッシュ関数処理回路130は、それぞれ個別の回路であり、並行してデータを処理することができる。以下に、CPU101、IPsec制御回路110

、暗号処理回路 120、およびハッシュ関数処理回路 130 のそれぞれの処理手順を説明する。

【0154】

図 20 は、データ暗号化時の各回路の動作を時系列で示す第 1 の図である。以下、図 20 に示す処理をステップ番号に沿って説明する。なお、第 1 の実施の形態では、暗号処理回路 120 は DES 処理を行いハッシュ関数処理回路 130 は、SHA1 処理を行う。

【0155】

[ステップ S31] 各回路が初期値を設定する。暗号処理回路 120 は、アルゴリズム (DES) と鍵を設定する。このときライト要求信号 (DREQ_WR[0]) はアサート中である。ハッシュ関数処理回路 130 は、アルゴリズム (SHA1) を設定する。このときライト要求信号 (DREQ_WR[1]) はアサート中である。IPsec 制御回路 110 は、処理対象データ (ターゲット) の取得を待機 (スタンバイ) している。このとき、CPU101 がメインのバスのバスマスタとなっている。

【0156】

[ステップ S32] IPsec 制御回路 110 は、DMA 転送に必要なパラメータとして、ソースアドレス、ディスティネーションアドレス、データレングスを設定する。また、暗号処理回路 120 は、スタンバイ状態となる。

【0157】

[ステップ S33] IPsec 制御回路 110 は、コントロール／モードレジスタ [1:0] 内の動作開始制御ビットをオン (値を「1」) にする。

[ステップ S34] IPsec 制御回路 110 は、外部バス解放要求信号 (BRQ) をアサートする。

【0158】

[ステップ S35] CPU101 は、外部バス開放要求受付信号 (BGNT) をアサートし、バスマスタではなくなる。IPsec 制御回路 110 は、バスマスタとなり、DMA 転送によりメモリ 103 から内蔵 RAM111 へ 64 バイトデータを取り込む。

【0159】

[ステップS36] IPsec制御回路110は、データ取り込みを終了し、外部バス解放要求信号(BRQ)をディアサートする。CPU101は、外部バス開放要求受付信号(BGNT)をディアサートし、バスマスタとなる。

【0160】

[ステップS37] IPsec制御回路110は、暗号処理回路120へ64ビットのデータを2回に分けて書き込み、書き込みを終了する。このとき、IPsec制御回路110は、データレングスレジスタから8バイト分差し引く。暗号処理回路120は、IPsec制御回路110からのデータ書き込み終了と同時に、暗号処理（暗号化または復号）を開始する。このとき、暗号処理回路120は、ライト要求信号(DREQ_WR[0])をディアサートする。この間、CPU101は、内部バスを使用して他の処理（パケット受信やプロトコルの処理）を実行する。

【0161】

[ステップS38] 暗号処理回路120は、暗号処理を実行する。IPsec制御回路110は、外部バス解放要求信号(BRQ)をアサートする。この時点では、CPU101が継続してバスマスタである。

【0162】

[ステップS39] CPU101は、外部バス開放要求受付信号(BGNT)をアサートし、バスマスタではなくなる。IPsec制御回路110は、バスマスタとなり、DMA転送によりメモリ103から内蔵RAM111へ64バイトデータを取り込む。

【0163】

図21は、データ暗号化時の各回路の動作を時系列で示す第2の図である。以下、図21に示す処理をステップ番号に沿って説明する。

[ステップS40] IPsec制御回路110は、データ取り込みを終了し、外部バス解放要求信号(BRQ)をディアサートする。CPU101は、外部バス開放要求受付信号(BGNT)をディアサートし、バスマスタとなる。この間に暗号処理回路120における暗号処理が終了し、暗号処理回路120によりリード要求信

号(DREQ_RD)がアサートされる。

【0164】

[ステップS41] IPsec制御回路110は、暗号処理回路120からのデータ転送開始を認識し、64ビットのデータを2回に分けて取り込む。この間、CPU101は、内部バスを使用して他の処理(パケット受信やプロトコルの処理)を実行する。

【0165】

[ステップS42] IPsec制御回路110は、外部バス解放要求信号(BRQ)をアサートする。

[ステップS43] CPU101は、外部バス開放要求受付信号(BGNT)をアサートし、バスマスタではなくなる。IPsec制御回路110は、バスマスタとなり、DMA転送により内蔵RAM111からメモリ103へ64バイトデータを書き出す。

【0166】

[ステップS44] IPsec制御回路110は、データ書き出しを終了し、外部バス解放要求信号(BRQ)をデアサートする。CPU101は、外部バス開放要求受付信号(BGNT)をデアサートし、バスマスタとなる。暗号処理回路120は、リード要求信号(DREQ_RD)をデアサートすると共にライト要求信号(DREQ_WR[0])をアサートし、スタンバイ状態となる。

【0167】

[ステップS45] 上記ステップS34～ステップS44の処理を、データレングスレジスタが0になるまで繰り返し実行する。データレングスレジスタが0になると、ステップS46の処理に進められる。

【0168】

[ステップS46] IPsec制御回路110は、処理終了割り込み信号が出力される。CPU101は、メモリ103にデータレングス長分の暗号処理結果を格納する。

【0169】

次に、ハッシュ関数に基づくハッシュ値生成処理について説明する。

図 22 は、ハッシュ値生成処理時の各回路の動作を時系列で示す図である。以下、図 22 に示す処理をステップ番号に沿って説明する。

【0170】

〔ステップ S51〕 各回路が初期値を設定する。暗号処理回路 120 は、アルゴリズム (DES) と鍵を設定する。このときライト要求信号 (DREQ_WR[0]) はアサート中である。ハッシュ関数処理回路 130 は、アルゴリズム (SHA1) を設定する。このときライト要求信号 (DREQ_WR[1]) はアサート中である。IPsec 制御回路 110 は、処理対象データ (ターゲット) の取得を待機 (スタンバイ) している。このとき、CPU101 がメインのバスのバスマスタとなっている。

【0171】

〔ステップ S52〕 IPsec 制御回路 110 は、DMA 転送に必要なパラメータとして、ソースアドレス、ディスティネーションアドレス、データレングスを設定する。

【0172】

〔ステップ S53〕 IPsec 制御回路 110 は、コントロール/モードレジスタ [1:0] 内の動作開始制御ビットをオン (値を「1」) にする。

〔ステップ S54〕 IPsec 制御回路 110 は、外部バス解放要求信号 (BRQ) をアサートする。

【0173】

〔ステップ S55〕 CPU101 は、外部バス開放要求受付信号 (BGNT) をアサートし、バスマスタではなくなる。IPsec 制御回路 110 は、バスマスタとなり、DMA 転送によりメモリ 103 から内蔵 RAM111 へ 128 バイトデータを取り込む。

【0174】

〔ステップ S56〕 IPsec 制御回路 110 は、データ取り込みを終了し、外部バス解放要求信号 (BRQ) をデリアサートする。CPU101 は、外部バス開放要求受付信号 (BGNT) をデリアサートし、バスマスタとなる。

【0175】

【ステップS57】IPsec制御回路110は、ハッシュ関数処理回路130へ128ビットのデータを16回に分けて書き込み、書き込みを終了する。このとき、IPsec制御回路110は、データレングスレジスタから64バイト分差し引く。ハッシュ関数処理回路130は、IPsec制御回路110からのデータ書き込み終了と同時に、ハッシュ関数処理を開始する。このとき、ハッシュ関数処理回路130は、ライト要求信号(DREQ_WR[1])をデリアサートする。この間、CPU101は、内部バスを使用して他の処理（パケット受信やプロトコルの処理）を実行する。

【0176】

【ステップS58】上記ステップS54～ステップS57の処理を、データレングスレジスタが0になるまで繰り返し実行する。データレングスレジスタが0になると、ステップS59の処理に進められる。

【0177】

【ステップS59】IPsec制御回路110は、処理終了割り込み信号が出力される。CPU101は、生成されたハッシュ値を読み取る。

次に、暗号処理およびハッシュ関数処理を行うときの動作波形を、タイミングチャートを参照して説明する。

【0178】

図23は、IPsec制御回路と暗号処理回路との間のバスの動作波形を示す第1のタイミングチャートである。図23では、上段にIPsec制御回路110側の端子における信号が示されており、下段に暗号処理回路120側の端子における信号および暗号処理回路120の内部データが示されている。

【0179】

IPsec制御回路110の信号としては、アドレス信号(IP_A)、ライトストロブ信号(IP_WRX[0])、リードストロブ信号(IP_RDX)、ライトデータバス(IP_WD[31:0])、リードデータバス(IP_RD[31:0])、ライト要求信号(DREQ_WR)、およびリード要求信号(DREQ_RD)が示されている。

【0180】

暗号処理回路120側の信号は、アドレス信号(IP_A)、ライトストロブ信

号(IP_WRX[0])、リードストローブ信号(IP_RDX)、ライトデータバス(IP_WD[31:0])、リードデータバス(IP_RD[31:0])、第1データレジスタ値(DES DR_U)、第2データレジスタ値(DES DR_L)、演算開始信号(des_start)、演算ステータス(DSTA)、ライト要求信号(DREQ_WR)、およびリード要求信号(DREQ_RD)が示されている。第1データレジスタ値(DES DR_U)は、64ビットデータレジスタ群124内的一方のデータレジスタ(第1データレジスタ)の値である。第2データレジスタ値(DES DR_L)は、64ビットデータレジスタ群124内の他方のデータレジスタ(第2データレジスタ)の値である。

【0181】

なお、図23に示す各信号は、ローアクティブの信号である。

ライト信号でIPsec制御回路110からのデータが第1データレジスタと第2データレジスタに32ビットずつ書き込まれる。レジスタにデータが書き込まれると、演算開始信号(des_start)がアサートされる。このとき、演算中ステータス(DSTA)もオンとなる。演算終了は16進カウンタ122で終了回数を示すときに制御信号が出され、演算中ステータス(DSTA)がオフとなる。

【0182】

アドレス信号(IP_A)は、第1データレジスタと第2データレジスタとを認識するためのアドレスの役割をする1ビットの信号で、第1データレジスタに書き込む場合にはアドレス信号(IP_A)は「0」(ローレベル)、第2データレジスタに書き込む場合にはアドレス信号(IP_A)は「1」(ハイレベル)を示す。

【0183】

ライト要求信号(DREQ_WR)は、暗号処理回路120からIPsec制御回路110にデータリクエストを示す信号で演算中でない、演算結果リード待ち状態でもないときにアサートされる。まず演算前にライト要求信号(DREQ_WR)がアサートされていて、64ビットデータレジスタ群124へのデータ書き込み終了後にディアサートされる。演算終了後、リード要求信号(DREQ_RD)がアサートされる。リード要求信号(DREQ_RD)は、データリードリクエスト信号であり、64ビットデータレジスタ群124に演算結果が上書きされた状態で、そのデータがリードされるのを待っていることを示している。IPsec制御回路110がリード

を行った後に、リード要求信号(DREQ_RD)はディアサートされ、ライト要求信号(DREQ_WR)が再びアサートされる。また、暗号回路の場合はIPsec制御回路110が演算対象データ書き込み、読み出し両方を行うため、IPsec制御回路110から暗号回路へライトデータを運ぶライトデータバス(IP_WD[31:0])と、暗号回路からIPsec制御回路110へリードデータを運ぶリードデータバス(IP_RD[31:0])とが設けられている。

【0184】

以下に、図23に示す信号の変化を時系列で詳細に説明する。

時刻t1にライト要求信号(DREQ_WR)がアサートされる。

時刻t2(時刻t1の1周期後)に、ライトストロブ信号(IP_WRX[0])がアサートされる。同時に、ライトデータバス(IP_WD[31:0])に対して、IPsec制御回路110から処理対象データ(32ビット)が出力される。

【0185】

時刻t3(時刻t2の1周期後)に、処理対象データが暗号処理回路120内の第1データレジスタ値(DESDR_U)に格納される。

時刻t4(時刻t3の1周期後)に、ライトデータバス(IP_WD[31:0])への処理対象データの出力が停止され、ライトストロブ信号(IP_WRX[0])がディアサートされる。同時に、IPsec制御回路110によりアドレス信号(IP_A)が「1」に設定される。すなわち、第2データレジスタのアドレスが指定される。

【0186】

時刻t5(時刻t4の1周期後)に、ライトストロブ信号(IP_WRX[0])がアサートされる。同時に、ライトデータバス(IP_WD[31:0])に対して、IPsec制御回路110から処理対象データ(32ビット)が出力される。

【0187】

時刻t6(時刻t5の1周期後)に、処理対象データが暗号処理回路120内の第2データレジスタ値(DESDR_L)に格納される。

時刻t7(時刻t6の1周期後)に、IPsec制御回路110によりアドレス信号(IP_A)が「0」に設定され、ライトストロブ信号(IP_WRX[0])がディアサートされる。同時に、ライトデータバス(IP_WD[31:0])へのデータ出力も停

止する。また、暗号処理回路 120 内部では、演算開始信号(des__start)がアサートされる。

【0188】

時刻 t 8 (時刻 t 7 の 1 周期後) に、演算中ステータス(DSTA)がオンとなり、演算開始信号(des__start)がデリアサートされる。

時刻 t 9 (時刻 t 8 の 1 周期後) に、ライト要求信号(DREQ__WR)がネゲートされる。その後、暗号処理回路 120 内で DES 処理が実行される。

【0189】

図 24 は、IPsec 制御回路と暗号処理回路との間のバスの動作波形を示す第 2 のタイミングチャートである。図 24 は、DES 処理完了後の動作波形を示している。

【0190】

DES 処理が完了した時刻 t 11 に、演算中ステータス(DSTA)がオフとなる。

時刻 t 12 (時刻 t 11 の 1 周期後) に、暗号処理回路 120 によりリード要求信号(DREQ__RD)がアサートされる。

【0191】

時刻 t 13 (時刻 t 12 の 1 周期後) に、リードストロブ信号(IP__RDX)がアサートされる。

時刻 t 14 (時刻 t 13 の 1 周期後) に、リードデータバス(IP__RD[31:0])に対して、暗号処理回路 120 から処理結果データ (32 ビット) が出力され、IPsec 制御回路 110 で読み込まれる。このとき、アドレス信号(IP__A)が「0」であるため、リードデータバス(IP__RD[31:0])には、第 1 データレジスタ内のデータが出力される。

【0192】

時刻 t 15 (時刻 t 14 の 1 周期後) に、リードストロブ信号(IP__RDX)がデリアサートされ、IPsec 制御回路 110 によりアドレス信号(IP__A)が「1」に設定される。

【0193】

時刻 t 16 (時刻 t 15 の 1 周期後) に、リードストロブ信号(IP__RDX)が

アサートされる。同時に、リードデータバス(IP_RD[31:0])へのデータ出力が一旦停止する。

【0194】

時刻 t 17 (時刻 t 16 の 1 周期後) に、リードデータバス(IP_RD[31:0])に対して、暗号処理回路 120 から処理結果データ (32 ビット) が出力され、IPsec 制御回路 110 で読み込まれる。このとき、アドレス信号(IP_A)が「1」であるため、リードデータバス(IP_RD[31:0])には、第 2 データレジスタ内のデータが出力される。

【0195】

時刻 t 18 (時刻 t 17 の 1 周期後) に、リードストロブ信号(IP_RDX)がデリアサートされ、IPsec 制御回路 110 によりアドレス信号(IP_A)が「1」に設定される。

【0196】

時刻 t 19 (時刻 t 18 の 1 周期後) に、リード要求信号(DREQ_RD)がデリアサートされる。同時に、リードデータバス(IP_RD[31:0])へのデータ出力が停止する。

【0197】

時刻 t 20 (時刻 t 19 の 1 周期後) に、ライト要求信号(DREQ_WR)がアサートされ、続くデータの DES 処理が行われる。

図 25 は、IPsec 制御回路と暗号処理回路との間のバスの動作波形を示すタイミングチャートである。図 25 では、上段に IPsec 制御回路 110 側の端子における信号が示されており、下段に暗号処理回路 120 側の端子における信号および暗号処理回路 120 の内部データが示されている。なお、この例は、ハッシュ関数処理回路 130 による SHA1 処理に関する動作波形である。

【0198】

IPsec 制御回路 110 の信号としては、ライトストロブ信号(IP_WRX[1])、ライトデータバス(IP_WD[31:0])、およびライト要求信号(DREQ_WR)が示されている。

【0199】

ハッシュ関数処理回路130の信号としては、ライトストロブ信号(IP_WRX[1])、ライトデータバス(IP_WD[31:0])、第1データレジスタ値(W0[31:0])、第16データレジスタ値(W15[31:0])、演算開始信号(hash_start)、演算中ステータス(HSTA)、およびライト要求信号(DREQ_WR)が示されている。

【0200】

ライトストロブ信号(IP_WRX[1])に応じて、IPsec制御回路110からのデータが512ビットデータレジスタ群134を構成する16個のデータレジスタに書き込まれる。全16個のデータレジスタにデータが書き込まれると、データセット完了と認識され、演算開始信号(hash_start)がアサートされ、演算中ステータス(HSTA)がオンに設定される。演算終了時は、80進カウンタ132で終了回数を示したとき演算終了信号が出力され、演算中ステータス(HSTA)がオフとなる。

【0201】

なお、512ビットデータレジスタ群134を構成する各データレジスタはシフトレジスタで構成されている。すなわち、毎回、第1データレジスタに書き込みが行われ、ライトストロブ信号が入力される毎に各データレジスタの値がそれぞれ次の段のデータレジスタにシフトされる。そのため、512ビットデータレジスタ群134内の任意のデータレジスタのアドレスを識別するための信号は不要となる。

【0202】

なお、ハッシュ関数処理回路130からIPsec制御回路110へのデータリクエストを示すデータ要求信号(DREQ_WR)は、ハッシュ関数処理回路130が演算中でないときにアサートされる。具体的には、まず演算前にデータ要求信号(DREQ_WR)がアサートされている。そして、512ビットデータレジスタへのデータ書き込み終了後にディアサートされる。演算が終了すると、再びデータ要求信号(DREQ_WR)がアサートされる。

【0203】

ところで、ハッシュ関数処理回路130内部には、図17に示すように演算結果格納用のハッシュ値格納レジスタ131と、ハッシュ演算用の512ビットデ

ータレジスタ群 134 とが存在する。ハッシュ関数はアルゴリズムの構成上、処理単位を 512 ビット倍数にパディングし、必ず処理単位は最低一回以上の何回かのハッシュ関数処理を重ねて終了する。

【0204】

最初はアルゴリズム固有の定数をハッシュ値として演算に使用し、二回目以降前の演算結果を演算途中結果と加算して演算結果となり、ハッシュ関数処理回路内部レジスタに格納される。パケット認証の場合、1 パケット分の処理が終わればハッシュ処理は終了で、最後にハッシュ関数内部のハッシュ値格納レジスタをリードすればよい。IPsec 制御回路 110 は演算対象データ書き込みのみを行うため、IPsec 制御回路 110 からハッシュ関数処理回路 130 へライトデータを運ぶ専用のライトデータバス(IP_WD[31:0])が設けられている。

【0205】

以下に、図 25 に示す信号の変化を時系列で詳細に説明する。

まず、時刻 t31 にデータ要求信号(DREQ_WR)がデassertされる。

時刻 t32 (時刻 t31 の 1 周期後) に、ライトストロブ信号(IP_WRX[0])がアassertされると共に、IPsec 制御回路 110 によりライトデータバス(IP_WD[31:0])に処理対象データが出力される。

【0206】

時刻 t33 (時刻 t32 の 1 周期後) に、ライトデータバス(IP_WD[31:0])を介して出力された処理対象データが第 1 データレジスタ(W0[31:0])に設定される。

【0207】

時刻 t34 (時刻 t33 の 1 周期後) に、ライトストロブ信号(IP_WRX[0])がデassertされると共に、ライトデータバス(IP_WD[31:0])への処理対象データの出力が停止する。

【0208】

その後、データが繰り返し転送される。16 回のデータ転送が行われる。

16 回目のデータ転送開始の時刻 t41 に、ライトストロブ信号(IP_WRX[0])がアassertされると共に、IPsec 制御回路 110 によりライトデータバス

(IP_WD[31:0])に処理対象データが出力される。

【0209】

時刻 t 4 2 (時刻 t 4 1 の 1 周期後) に、ライトデータバス (IP_WD[31:0]) を介して出力された処理対象データが第 1 データレジスタ (W0[31:0]) に設定される。同時に、512 ビットデータレジスタ群 134 内のデータが順次シフトされ、第 16 データレジスタ値 (W15[31:0]) に、最初に転送された 32 ビットのデータが格納される。

【0210】

時刻 t 4 3 (時刻 t 4 2 の 1 周期後) に、ライトストロブ信号 (IP_WRX[1]) がディアサートされると共に、ライトデータバス (IP_WD[31:0]) への処理対象データの出力が停止する。このとき、16 回のデータ書き込み完了が認識され、演算開始信号 (hash_start) がアサートされる。

【0211】

時刻 t 4 4 (時刻 t 4 3 の 1 周期後) に、演算ステータス (HSTA) がオンに設定され、演算開始信号 (hash_start) がディアサートされる。

時刻 t 4 5 (時刻 t 4 4 の 1 周期後) に、データ要求信号 (DREQ_WR) がアサートされ、ハッシュ関数処理が実行される。

【0212】

演算回数が 80 回に達するとハッシュ関数処理完了と判断され、その時刻 t 5 1 に演算ステータス (HSTA) がオフに設定される。

時刻 t 5 2 (時刻 t 5 1 の 1 周期後) に、データ要求信号 (DREQ_WR) がディアサートされる。

【0213】

以上のように、セキュリティネットワークコントローラ 100 によれば、IPsec 制御回路 110 が暗号処理回路 120 やハッシュ関数処理回路 130 へのデータの入出力を行うため、CPU 101 の処理負荷が減少する。

【0214】

なお、通信データの暗号化の範囲や認証範囲は、通信プロトコルによって異なる。

図26は、トランスポートモードESPの暗号化範囲と認証範囲とを示す図である。図26(A)はIPv4のパケットを示し、図26(B)はIPv6のパケットを示す。

【0215】

IPv4のパケット50は、IPヘッダ51、ESPヘッダ52、TCPヘッダ53、データ54、ESPトレーラ55、ESP認証ヘッダ56で構成される。そのうち、TCPヘッダ53、データ54、ESPトレーラ55が暗号化範囲である。また、ESPヘッダ52、TCPヘッダ53、データ54、ESPトレーラ55が認証範囲である。

【0216】

IPv6のパケット60は、IPv6ヘッダ61、経路制御ヘッダ62、ESPヘッダ63、終点オプションヘッダ64、TCPヘッダ65、データ66、ESPトレーラ67、ESP認証ヘッダ68で構成される。そのうち、終点オプションヘッダ64、TCPヘッダ65、データ66、ESPトレーラ67が暗号化範囲である。また、ESPヘッダ63、終点オプションヘッダ64、TCPヘッダ65、データ66、ESPトレーラ67が認証範囲である。

【0217】

たとえばトランスポートモードESPを用いる際のパケット生成時には、範囲の情報に関して、鍵付きハッシュ関数(HMAC-SHA1, HMAC-MD5)を用いた認証処理が行われる。

【0218】

図27は、AHの認証範囲を示す図である。AHのパケット70の構成は、IPv4、IPv6、共に同じである。AHのパケット70は、IPヘッダ71、AHヘッダ72、TCPヘッダ73、データ74で構成される。AHのパケット70では、全てが認証範囲となる。すなわち、パケット全体の鍵付きハッシュ関数(HMAC-SHA1, HMAC-MD5)を用いた認証処理が必要となる。

【0219】

また、最大パケットサイズはネットワークの接続媒体によって決定される。たとえば、ネットワークの接続物理媒体としてイーサネット（登録商標）と呼ばれ

るネットワーク(IEEE802.3)を用いた場合の I P パケットのサイズを説明する。

【 0 2 2 0 】

なお、IEEE802.3のネットワークの一例として、1パケットの認証処理を例に発明の作用を説明する。認証処理においては、最初と最後にHMAC処理を行うため機能的には、CPUにて設定（HMAC処理かどうか、MD5／SHA1かどうかなどのモード設定）を行った後、シームレスにハードウェア処理が行われる。これにより、CPU負荷を軽減し、またハードウェアの持つパフォーマンス値を引き出すことが可能となる。また、DMAC(Direct Memory Access Controller)を用いてCPUにおいて処理を行う認証処理を独立して実行することもできる。

【 0 2 2 1 】

図 2 8 は、I P パケットサイズを示す図である。図 2 8 に示すように、I P パケット 8 0 は、I P v 6 ヘッダ 8 1、経路制御ヘッダ 8 2、終点オプションヘッダ 8 3、T C P ヘッダ 8 4、データ 8 5で構成される。このような構成のIEEE802.3に従ったネットワークの最大 I P パケットサイズは、約1500バイト（I P v 6 ヘッダ 4 0 バイト、その他1460バイト以下）である。

【 0 2 2 2 】

認証アルゴリズムであるハッシュ関数のMD5とSHA1は、64バイト(512ビット)ブロック関数である。従って、1パケットの処理でさえ、最大約20回を超えるハッシュ処理が行われる。また、鍵管理においてもハッシュ関数が使用される。

【 0 2 2 3 】

図 2 9 は、I K E メインモードを使用したフェーズ 1 を示す図である。フェーズ 1 は、鍵交換プロトコルの前半である。図 2 9 には、送信者(Intiator)と受信者(Responder)との間で交換されるメッセージが示されている。第 1 のメッセージ(Message#1)において、I S A K M P - S A (Source Address)のネゴシエーションが開始される。第 2 のメッセージ(Message#2)において、基本 S A が承認される。第 3 のメッセージ(Message#3)と第 4 のメッセージ(Message#4)とにより、互いの鍵が交換される。第 5 のメッセージ(Message#5)で、送信者の身元が受信

者に確認される。第6のメッセージ(Message#6)で、受信者の身元が送信者に確認される。なお、第5のメッセージ(Message#5)と第6のメッセージ(Message#6)とは、ペイロードが暗号化されている。

【0224】

IPv6で必携機能となるIPsecは、ネットワークを介して送りたいパケットを暗号・認証処理終了後、IPパケットを含むMACフレームに生成する。その後、通信インタフェースを経由して送信する。

【0225】

受信時には認証処理、復号化処理を行う。暗号処理、認証処理の範囲は、図26に示した通りであり、暗号化処理範囲は受信時には復号化処理範囲となる。認証とは、図に示した認証範囲データをハッシュ関数処理を施し、パケットの一番最後に付け加える処理である。

【0226】

IPパケットサイズの最大はネットワークの物理層としてIEEE802.3で規定されたイーサネット（登録商標）を用いる場合は、最大パケットサイズは約1500バイトである。つまり、ハッシュ関数による認証は最大約20回の処理を行い、暗号化に関しては、暗号アルゴリズムにDESや3DESを用いる場合、最大約180回の暗号処理を行う。

【0227】

また、図29に示した鍵交換プロトコルの前半であるフェーズ1の処理は、第1のメッセージから第6のメッセージまでの各メッセージの送受信で暗号処理が行われる。また、その後のフェーズ2でもSAを確立するためのセッションで多くの暗号処理、ハッシュ関数処理が行われる。たとえば、第3のメッセージと第4のメッセージでは、鍵生成のための値を交換するが、どちらも鍵生成に必要な交換値をハッシュ関数や、暗号アルゴリズムを用いて作成する。

【0228】

以上のように、暗号処理、ハッシュ関数処理は実にIPsecについて多用される。そのため、第1の実施の形態に示すようにCPUを介さないで、暗号処理やハッシュ処理（認証処理）の高速処理を実現できるようにすることによる装置

全体への処理能力の向上効果が高くなる。しかも、セキュリティネットワークコントローラへの I P s e c 制御回路の実装は、C P U の高速化に比べ安価に実現することができ、快適な通信速度を市場ニーズに合った価格で提供することが可能となる。

【0229】

図30は、従来技術と第1の実施の形態とにおける暗号処理の性能評価結果を示す図である。これは1496バイトデータの3DES-CBC暗号処理を行った場合の例である。図30の例では、ソフトウェア的に実行した場合（暗号化手順を記述したプログラムをC P U が実行する）、C P U と暗号処理回路との組み合わせで実行した場合（C P U がデータの入出力等を制御する）、および I P s e c 制御回路と暗号処理回路との組み合わせで実行した場合（第1の実施の形態に係る構成）を比較している。

【0230】

なお、ソフトウェア的な処理は、具体的には、処理手順をC言語でコード化し、フラッシュメモリに記憶させ、C P U に実行させている。また、I P s e c 制御回路+暗号処理回路の構成は、P L D (Programmable Logic Device) で実現したものである。

【0231】

ソフトウェア的に実行した場合、暗号化に264917 μ 秒、復号に264919 μ 秒を要している。C P U +暗号処理回路で実行した場合、暗号化に2977 μ 秒、復号に2979 μ 秒を要している。I P s e c 制御回路+暗号処理回路で実行した場合、暗号化に579 μ 秒、復号に581 μ 秒を要している。

【0232】

図31は、従来技術と第1の実施の形態とにおけるハッシュ関数処理の性能評価結果を示す図である。これは1500バイトデータのHMAC-SHA1ハッシュ関数処理を行った場合の例である。図31の例では、ソフトウェア的に実行した場合、C P U と暗号処理回路との組み合わせで実行した場合、および I P s e c 制御回路と暗号処理回路との組み合わせで実行した場合を比較している。なお、I P s e c 制御回路+暗号処理回路の構成は、P L D (Programmable Logic

Device)で実現したものである。

【0233】

ソフトウェア的に実行した場合、ハッシュ関数処理における暗号化に41309 μ 秒を要している。CPU+ハッシュ処理回路で実行した場合、ハッシュ関数処理における暗号化に2258 μ 秒を要している。IPsec制御回路+ハッシュ処理回路で実行した場合、ハッシュ関数処理における暗号化に297 μ 秒を要している。

【0234】

このように、実装評価として3DES-CBC暗号処理と、HMAC-SHA1処理とに関する処理性能を評価した。その結果、3DES-CBC暗号処理では、第1の実施の形態を適用することで従来技術であるソフトウェアによる処理の約457倍の高速化が達成されている。また、第1の実施の形態を適用することで、専用暗号回路とCPUとで処理を行わせた場合の約5倍の高速化が達成されている。

【0235】

また、HMAC-SHA1ハッシュ関数処理では、第1の実施の形態を適用することで、ソフトウェアによる処理の約139倍の高速化が達成されている。また、第1の実施の形態を適用することで、専用ハッシュ関数処理回路とCPUとにより処理を行わせた場合の約8倍の高速化が達成されている。

【0236】

このように、第1の実施の形態によれば、安全なデータ通信を非常に高速に行うことができる。したがって、動画データをストリーミング配信する場合であっても、セキュアなデータを安定して送受信することができる。

【0237】

しかも、暗号処理回路120とハッシュ関数処理回路130とは、入力されたデータの量を監視しており、所定量のデータの書き込みが行われると暗号処理や認証処理を自動的に開始するため、CPU等に負荷をかけずに処理を開始することができる。これは、ストリーミングのように処理対象のデータが連続して入力される場合に特に有効である。

【0238】

さらに、暗号処理回路 120 やハッシュ関数処理回路 130 はハードウェアマクロの処理開始タイミングを自動認識し、IPsec 制御回路 110 は暗号処理及び認証処理対象のデータを CPU 101 の代わりに専用のバスを介して間隔をあけずに供給する。これにより、暗号処理や認証処理がシームレスに実現され、暗号処理回路 120 とハッシュ関数処理回路 130 との処理能力を最大限に引き出すことができる。

【0239】

その結果、低パフォーマンス（たとえば、動作周波数が低い）の CPU を用いる場合でも、ストリーミングが考慮された高速処理を実現できる。低速の CPU を用いれば、低消費電力化や、低コストでの製造が容易となる。

【0240】

また、CPU 101 のワークメモリ領域を大量に占有しないため、CPU 101 による他の処理の処理効率が向上する。また、暗号処理や認証処理のための CPU 占有率が軽減されるため、システムとして機能させる際にも確実なパフォーマンスを保障することができる。また、IPsec を使用した快適な通信速度を持つネットワークサービスが可能となる。

【0241】

[第2の実施の形態]

第2の実施の形態は、暗号処理とハッシュ関数処理との並列処理を可能としたものである。

【0242】

図32は、第2の実施の形態のシステム構成例を示す図である。第2の実施の形態にかかるセキュリティネットワークコントローラ 200 は、CPU 201、通信インタフェース 202、メモリコントローラ 203、外部接続インタフェース 204、複数の IPsec 制御回路 211～214、複数の暗号処理回路 221～224、および複数のハッシュ関数処理回路 231～234 を有しており、これらの要素がバス 209 を介して接続されている。

【0243】

IPsec 制御回路 211 は、暗号処理回路 221 とハッシュ関数処理回路 2

31 とに、それぞれ専用のバスで接続されている。IPsec 制御回路 212 は、暗号処理回路 222 とハッシュ関数処理回路 232 とに、それぞれ専用のバスで接続されている。IPsec 制御回路 213 は、暗号処理回路 223 とハッシュ関数処理回路 233 とに、それぞれ専用のバスで接続されている。IPsec 制御回路 214 は、暗号処理回路 224 とハッシュ関数処理回路 234 とに、それぞれ専用のバスで接続されている。

【0244】

通信インタフェース 202 は、インターネット 91 を介して端末装置 92 に接続されている。メモリコントローラ 203 は、メモリ 205 に接続されている。外部接続インタフェース 204 は、カメラシステム 240 内のメイン CPU 241 に接続されている。メイン CPU 241 は、カメラシステム 240 内の他の回路 242, 243 やカメラ機構部 245 を制御する。

【0245】

このように、第 2 の実施の形態では、暗号処理および認証処理を行うための回路が多重化されている。図 32 の例では、それぞれの回路が 4 個ずつ設けられている。

【0246】

処理対象のデータは、複数の IPsec 制御回路 211 ~ 214 に分配される。各 IPsec 制御回路 211 ~ 214 は、分配されたデータの暗号処理またはハッシュ関数処理を制御する。

【0247】

このように回路を多重化することで、暗号処理やハッシュ関数処理対象のデータが大量である場合に、高速に処理することができる。また、転送速度の非常に高い性能が要求される場合（プリンタ画像転送など、500 ~ 1000 Mbps など）にも、回路を多重化させることで、所望の処理速度を得ることができる。なお、CPU 201 の処理性能を向上させれば処理性能は向上するが、CPU 201 の高速化は非常に高度な製造技術が必要となり、高価な回路になってしまう。図 32 のように、暗号処理やハッシュ関数処理の回路を多重化すれば、高いパフォーマンスを持たない CPU でも暗号処理やハッシュ関数処理を並行に行うことが

でき、1 パケット分の処理が高速化される。概算すると、多重化しない回路よりも4倍速くなる。

【0248】

なお、上記の各実施の形態では、カメラシステム10、240にセキュリティネットワークコントローラ100、200を実装した場合の例を示しているが、端末装置30、92側にも同様のセキュリティネットワークコントローラを実装することができる。

【0249】

また、カメラシステム10等の制御対象機器に対してセキュリティネットワークコントローラ100が外部接続されていてもよい。このようなシステムによって、入力手段を持たなくても、インターネット20を介したカメラシステム10の制御が可能となる。しかも、既存の回路を変更せずに、上記実施の形態に係るセキュリティネットワークコントローラを組み込むだけで、快適な通信速度で、安全にインターネットを介して様々な機器とネットワーク接続、IPsec機能を用いたセキュアで便利なサービスが実現できる。

【0250】

また、上記の実施の形態は、送受信するデータに関し、暗号処理（暗号化または復号）および認証処理（たとえば、ハッシュ関数処理）を行う場合の例であるが、暗号処理と認証処理との何れか一方の処理のみを行うセキュリティネットワークコントローラであってもよい。暗号処理のみを行うセキュリティネットワークコントローラにおいては、ハッシュ関数処理回路は不要となる。また、認証処理のみを行うセキュリティネットワークコントローラにおいては、暗号処理回路は不要となる。

【0251】

また、上記の実施の形態では、処理対象のデータを一旦メモリに格納し、そのメモリからDMA転送によりIPsec制御回路110がデータを取得しているが、IPsec制御回路110内に十分な容量の内蔵メモリが有る場合、処理対象のデータを通信インタフェース105等から直接IPsec制御回路110に格納することもできる。

【0252】

(付記1) データを保障するための処理を行うデータ保障装置において、
処理対象データを取得するデータ取得回路と、
入力されたデータの暗号処理を行う暗号処理回路と、
前記データ取得回路に第1のバスを介して接続されると共に前記暗号処理回路
に対して第2のバスを介して接続されており、前記データ取得回路が取得した前
記処理対象データを前記第1のバス経由で取得して内蔵メモリに格納し、前記処
理対象データを前記第2のバス経由で前記暗号処理回路に入力し、前記暗号処理
回路から前記第2のバス経由で暗号処理実行後の処理結果データを取得するデー
タ入出力制御回路と、
を有することを特徴とするデータ保障装置。

【0253】

(付記2) 前記データ入出力制御回路は、前記第1のバスのダイレクトメモ
リアクセスコントローラを有しており、前記処理対象データを前記データ取得回
路からダイレクトメモリアクセス転送によって取得することを特徴とする付記1
記載のデータ保障装置。

【0254】

(付記3) 前記第1のバスに接続され、前記データ取得回路が取得した前記
処理対象データを記憶する記憶装置をさらに有し、
前記データ入出力制御回路は、前記記憶装置から前記処理対象データを取得す
ることを特徴とする付記1記載のデータ保障装置。

【0255】

(付記4) 前記処理対象データを前記内蔵メモリに格納した後、前記処理対
象データを前記暗号処理の処理単位となる単位データ長に分割して、前記暗号処
理回路に入力することを特徴とする付記1記載のデータ保障装置。

【0256】

(付記5) 前記内部記憶装置は少なくとも2つの領域に分けられており、第
1の領域に格納された前記処理対象データが前記暗号処理回路で処理されている
間に、第2の領域に後続の前記処理対象データを格納することを特徴とする付記

4 記載のデータ保障装置。

【0257】

(付記6) 前記内蔵メモリは、暗号処理の前記単位データ長に相当する記憶容量単位の領域に分割されていることを特徴とする付記5記載のデータ保障装置。

【0258】

(付記7) 前記データ入出力制御回路は、ストリーミングで順次提供される前記処理対象データを、取得順に前記暗号処理回路に入力し、暗号処理が行われた処理結果データを取得する毎に出力することを特徴とする付記1記載のデータ保障装置。

【0259】

(付記8) 前記暗号処理回路は、入力された処理対象データの容量を検知し、所定の容量に達したときに前記処理対象データに対する暗号処理を実行することを特徴とする付記1記載のデータ保障装置。

【0260】

(付記9) 前記暗号処理回路が複数設けられ、複数の前記データ入出力制御回路が各前記暗号処理回路に個別に対応づけて接続されており、複数の前記データ入出力制御回路が前記処理対象データを分割して取得し、対応する前記暗号処理回路に対して並列に入力することを特徴とする付記1記載のデータ保障装置。

【0261】

(付記10) 入力されたデータの認証処理を行う認証処理回路を更に有し、前記データ入出力制御回路は前記認証処理回路に対して第3のバスを介して接続されており、暗号化対象である前記処理対象データを前記暗号処理に入力し、認証対象である前記処理対象データを前記認証処理回路に入力することを特徴とする付記1記載のデータ保障装置。

【0262】

(付記11) データを保障するための処理を行うデータ保障装置において、
処理対象データを取得するデータ取得回路と、
入力されたデータの認証処理を行う認証処理回路と、

前記データ取得回路に第1のバスを介して接続されると共に前記認証処理回路に対して第2のバスを介して接続されており、前記データ取得回路が取得した前記処理対象データを前記第1のバス経由で取得して内蔵メモリに格納し、前記処理対象データを前記第2のバス経由で前記認証処理回路に入力するデータ入出力制御回路と、

を有することを特徴とするデータ保障装置。

【0263】

(付記12) 保障されたデータをネットワークを介して送受信するデータ通信装置において、

送信データを生成するメインCPUと、

入力されたデータを暗号化する暗号処理回路と、

入力されたデータを前記ネットワークを介して送信する通信回路と、

前記メインCPUと前記通信回路に第1のバスを介して接続されると共に前記暗号処理回路に対して第2のバスを介して接続されており、前記メインCPUが取得した前記送信データを前記第1のバス経由で取得して内蔵メモリに格納し、前記送信データを前記第2のバス経由で前記暗号処理回路に入力し、前記暗号処理回路から前記第2のバス経由で暗号化後の暗号データを取得し、前記通信回路に対して入力するデータ入出力制御回路と、

を有することを特徴とするデータ通信装置。

【0264】

(付記13) 保障されたデータをネットワークを介して送受信するデータ通信装置において、

受信データを処理するメインCPUと、

入力されたデータを復号する暗号処理回路と、

前記ネットワークを介して送られた受信データを取得する通信回路と、

前記メインCPUと前記通信回路に第1のバスを介して接続されると共に前記暗号処理回路に対して第2のバスを介して接続されており、前記通信回路が取得した前記受信データを前記第1のバス経由で取得して内蔵メモリに格納し、前記受信データを前記第2のバス経由で前記暗号処理回路に入力し、前記暗号処理回

路から前記第2のバス経由で復号後の平文データを取得し、前記メインCPUに対して入力するデータ入出力制御回路と、

を有することを特徴とするデータ通信装置。

【0265】

(付記14) データを保障するためのデータ保障方法において、

データ取得回路で取得した処理対象データを、データ入出力制御回路が第1のバス経由で取得して内蔵メモリに格納し、

前記データ入出力制御回路が、前記処理対象データを第2のバス経由で暗号処理回路に入力し、

前記暗号処理回路が前記処理対象データの暗号処理を行い、

暗号処理実行後の処理結果データを、前記暗号処理回路から前記データ入出力制御回路に渡す、

ことを特徴とするデータ保障方法。

【0266】

(付記15) データを保障するためのデータ保障方法において、

データ取得回路で取得した処理対象データを、データ入出力制御回路が第1のバス経由で取得して内蔵メモリに格納し、

前記データ入出力制御回路が、前記処理対象データを第2のバス経由で認証処理回路に入力し、

前記認証処理回路が前記処理対象データの認証処理を行う、

ことを特徴とするデータ保障方法。

【0267】

【発明の効果】

以上説明したように本発明では、処理対象データを第1のバスを介して取得してデータ入出力制御回路の内蔵メモリに格納し、その処理対象データを、第2のバスを介して暗号処理回路に入力するようにしたため、一旦処理対象データをデータ入出力制御回路の内蔵メモリに格納した後は、CPUや第1のバスを介さずに暗号処理を行うことができ、システムを管理するCPUの処理負荷が軽減される。

【図面の簡単な説明】**【図 1】**

実施の形態に適用される発明の概念図である。

【図 2】

第 1 の実施の形態に係るシステム構成例を示す図である。

【図 3】

カメラ内部の回路構成を示す図である。

【図 4】

データ送信の手順を示すフローチャートである。

【図 5】

データ配信の第 1 のステップを示す図である。

【図 6】

データ配信の第 2 のステップを示す図である。

【図 7】

データ配信の第 3 のステップを示す図である。

【図 8】

データ配信の第 4 のステップを示す図である。

【図 9】

データ配信の第 5 のステップを示す図である。

【図 10】

データ配信の第 6 のステップを示す図である。

【図 11】

データ配信の第 7 のステップを示す図である。

【図 12】

データ配信の第 8 のステップを示す図である。

【図 13】

データの受信手順を示すフローチャートである。

【図 14】

セキュリティネットワークコントローラの内部構成例を示す図である。

【図 15】

I P s e c 制御回路の内蔵 R A M の D E S 処理時の構成を示す図である。

【図 16】

I P s e c 制御回路の内蔵 R A M の S H A 1 処理時の構成を示す図である。

【図 17】

I P s e c 制御回路の内部構成を示す図である。

【図 18】

暗号処理回路の内部構成を示す図である。

【図 19】

ハッシュ関数処理回路の内部構成を示す図である。

【図 20】

データ暗号化時の各回路の動作を時系列で示す第 1 の図である。

【図 21】

データ暗号化時の各回路の動作を時系列で示す第 2 の図である。

【図 22】

ハッシュ値生成処理時の各回路の動作を時系列で示す図である。

【図 23】

I P s e c 制御回路と暗号処理回路との間のバスの動作波形を示す第 1 のタイミングチャートである。

【図 24】

I P s e c 制御回路と暗号処理回路との間のバスの動作波形を示す第 2 のタイミングチャートである。

【図 25】

I P s e c 制御回路と暗号処理回路との間のバスの動作波形を示すタイミングチャートである。

【図 26】

トランスポートモード E S P の暗号化範囲と認証範囲とを示す図である。図 26 (A) は I P v 4 のパケットを示し、図 26 (B) は I P v 6 のパケットを示す。

【図 27】

AH の認証範囲を示す図である。

【図 28】

IP パケットサイズを示す図である。

【図 29】

IKE メインモードを使用したフェーズ 1 を示す図である。

【図 30】

従来技術と第 1 の実施の形態とにおける暗号処理の性能評価結果を示す図である。

【図 31】

従来技術と第 1 の実施の形態とにおけるハッシュ関数処理の性能評価結果を示す図である。

【図 32】

第 2 の実施の形態のシステム構成例を示す図である。

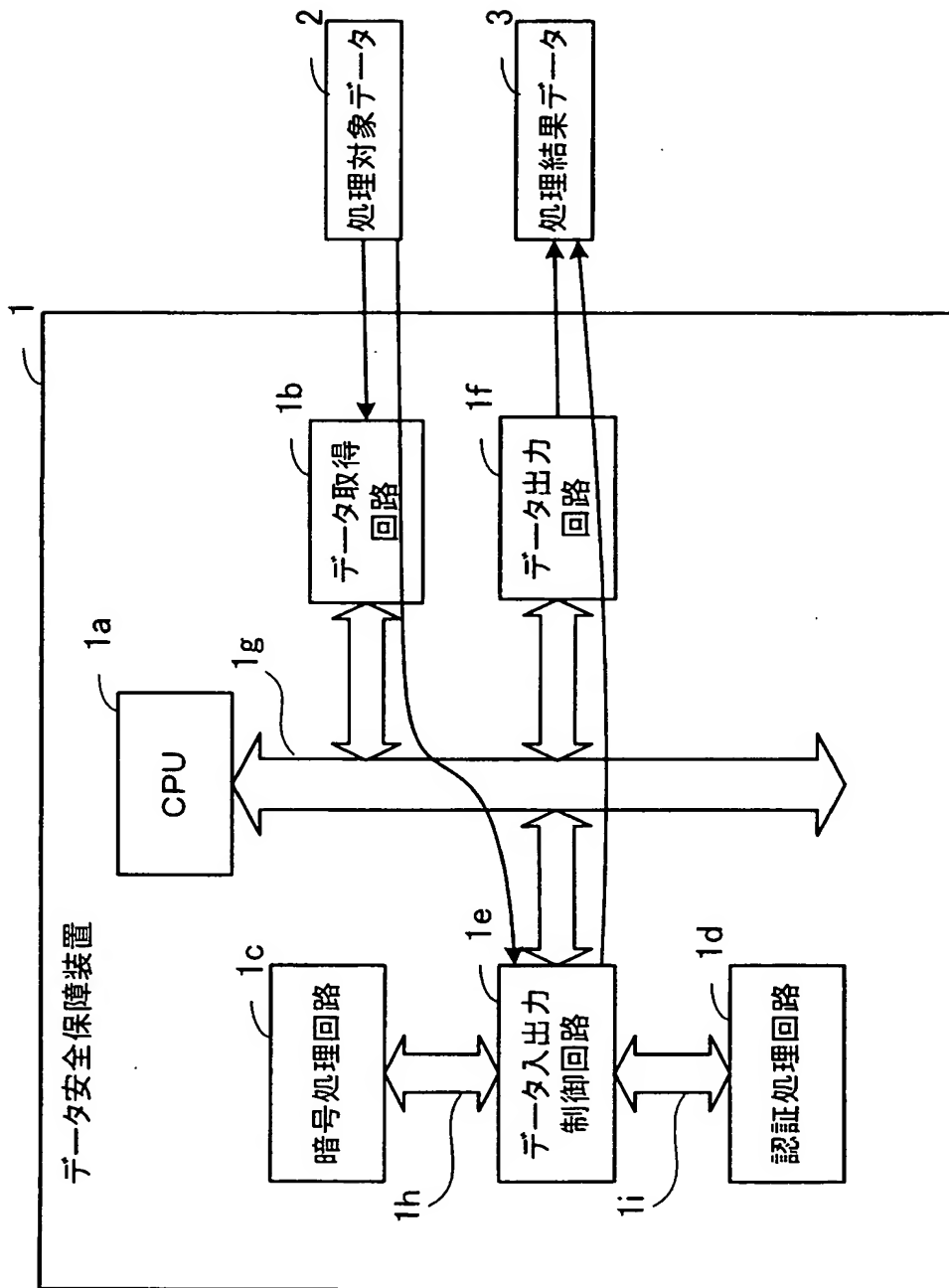
【符号の説明】

- 1 データ保障装置
 - 1 a CPU
 - 1 b データ取得回路
 - 1 c 暗号処理回路
 - 1 d 認証処理回路
 - 1 e データ入出力制御回路
 - 1 f データ出力回路
 - 1 g 第 1 のバス
 - 1 h 第 2 のバス
 - 1 i 第 3 のバス
- 2 処理対象データ
- 3 処理結果データ

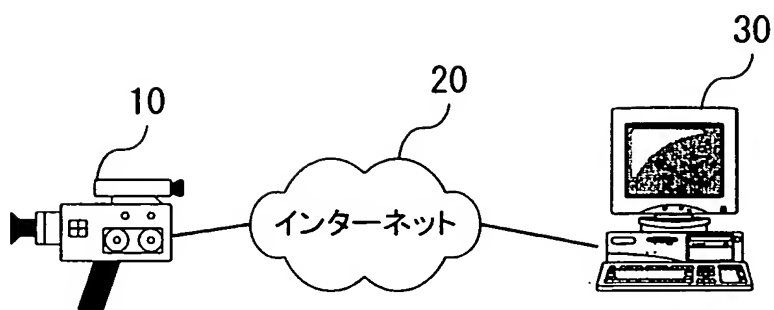
【書類名】

図面

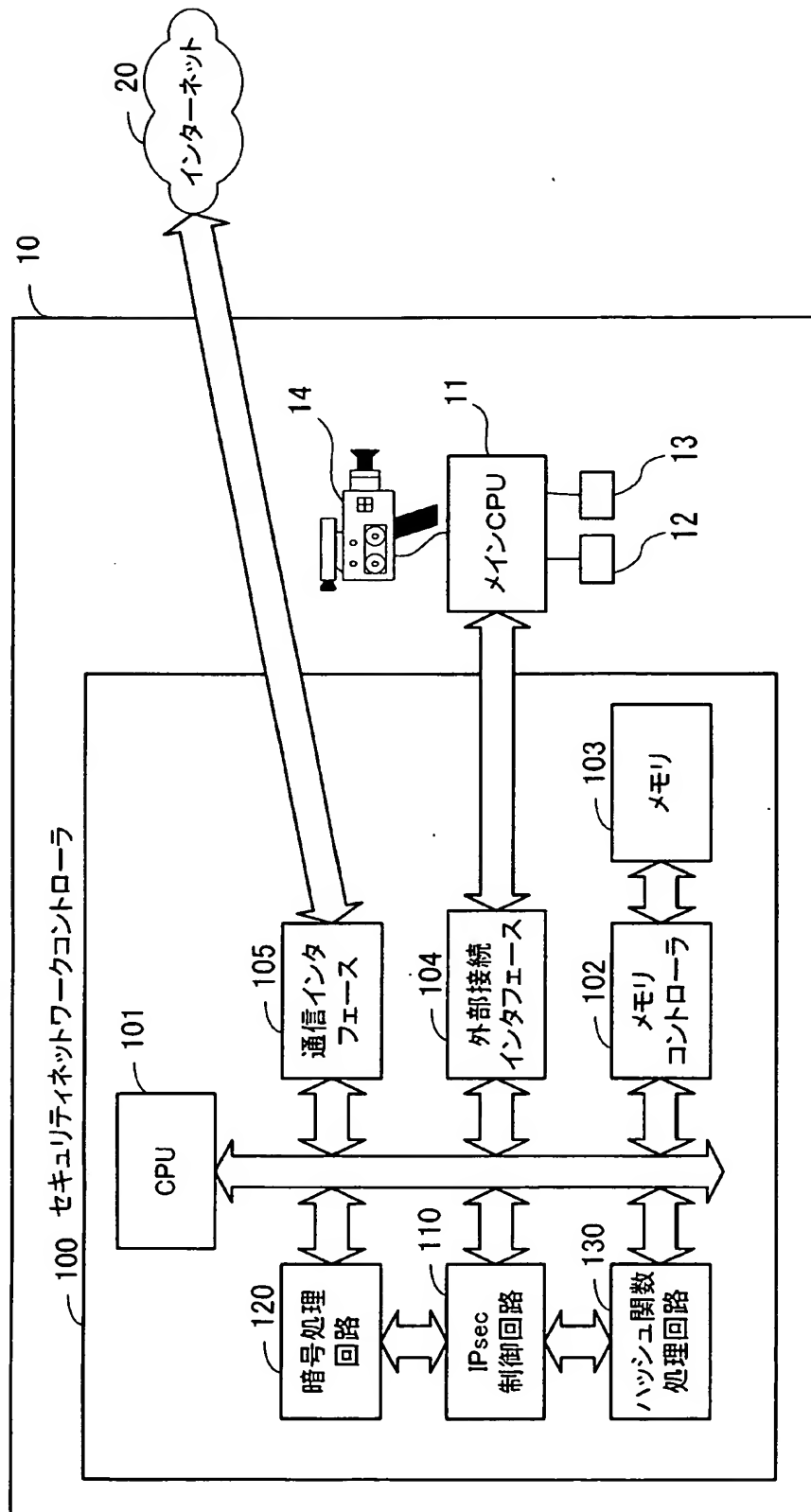
【図 1】



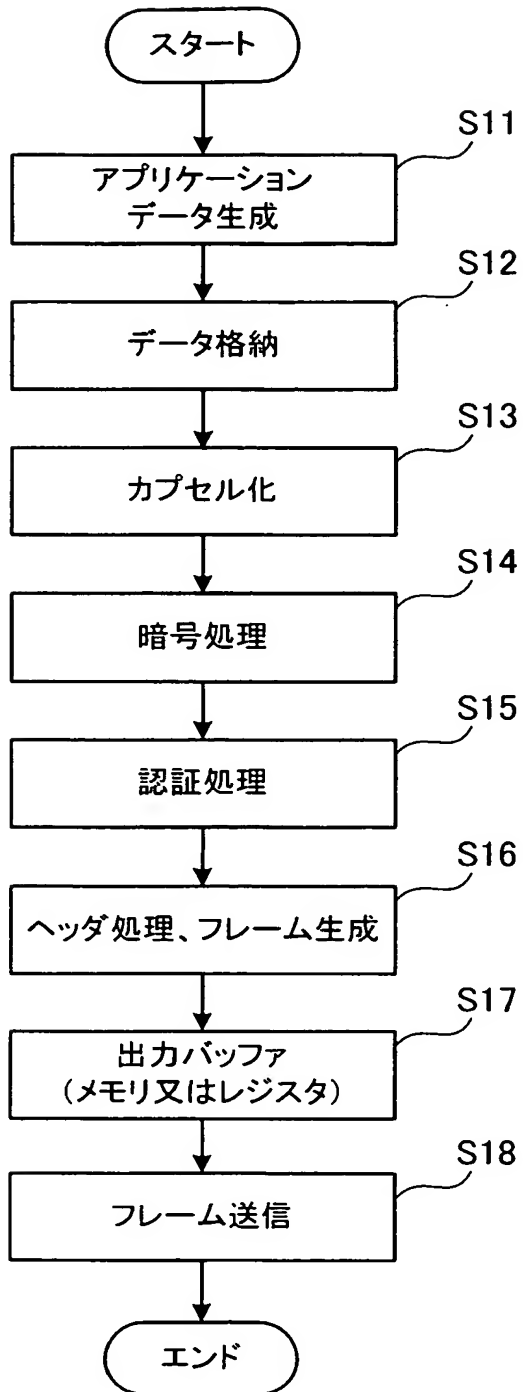
【図 2】



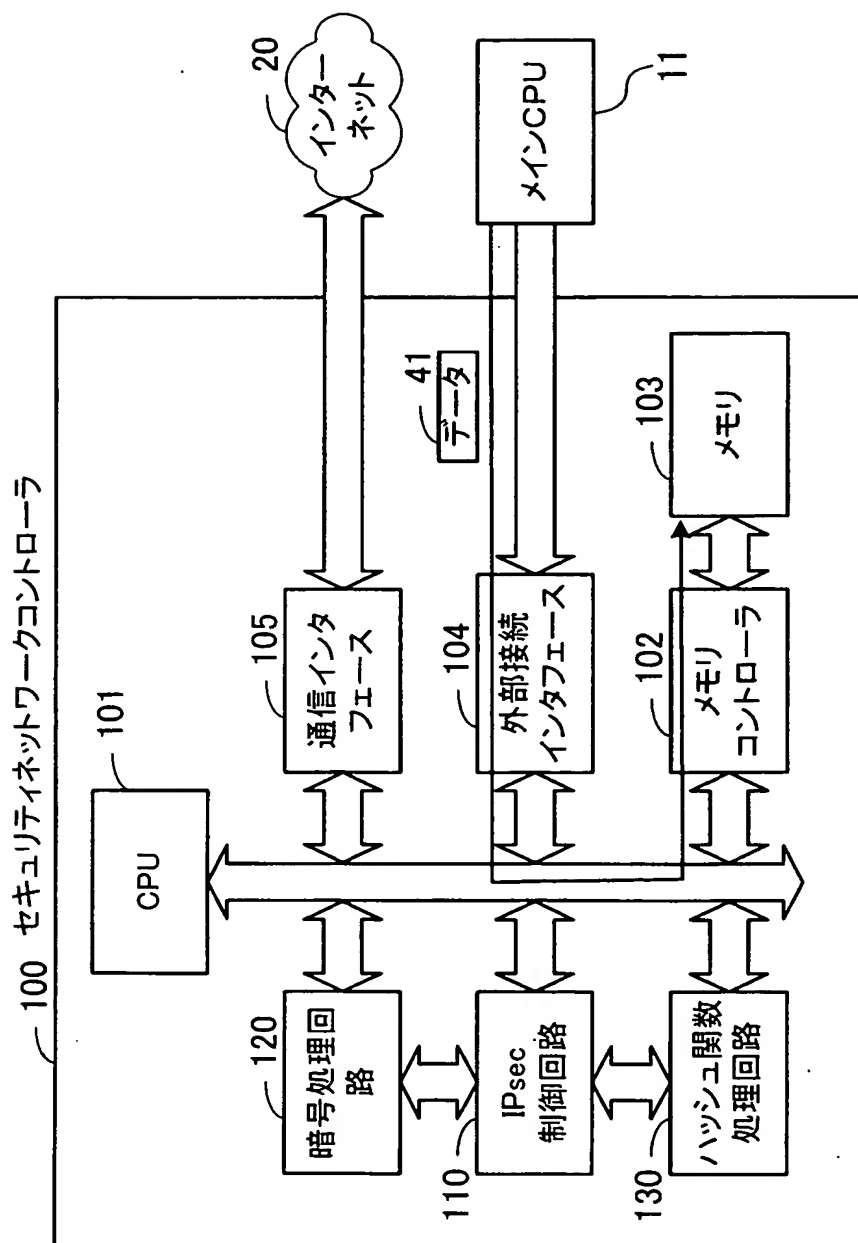
【図 3】



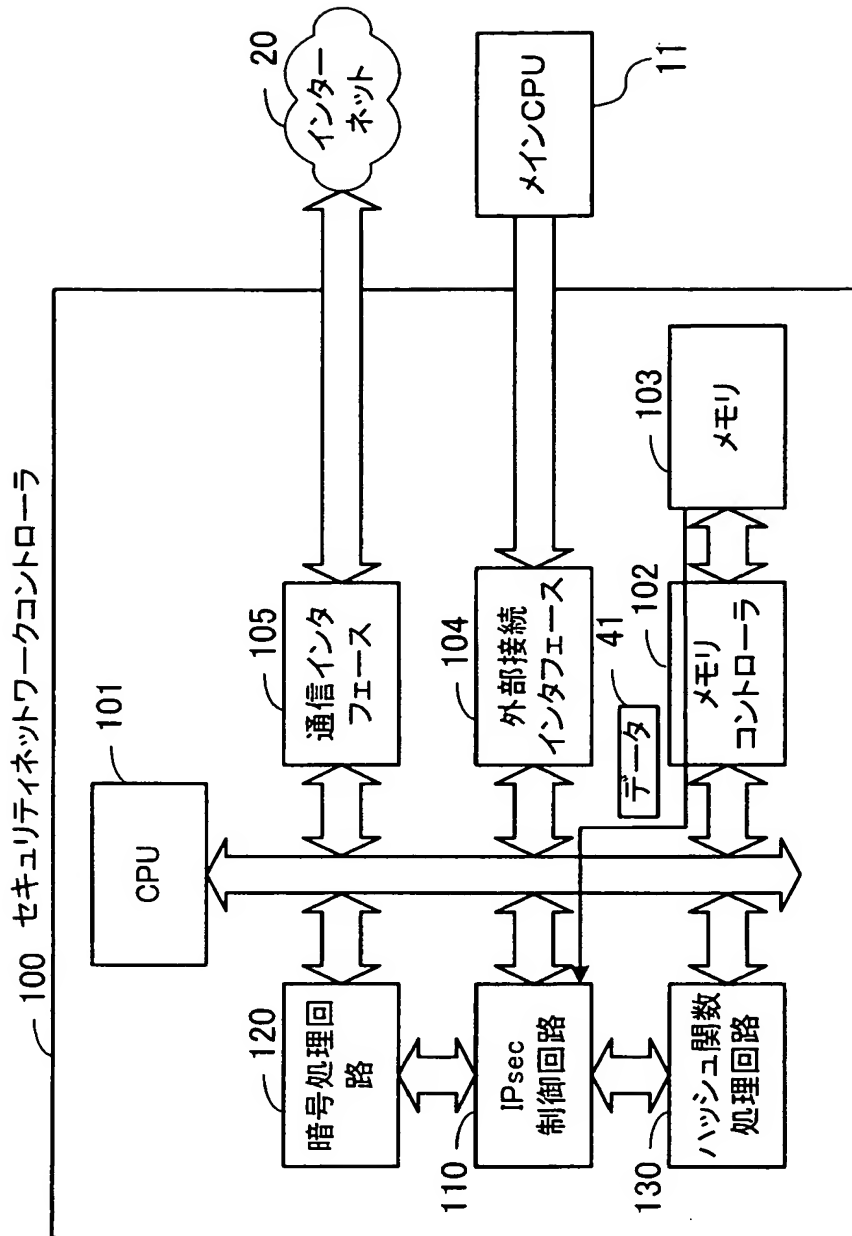
【図 4】



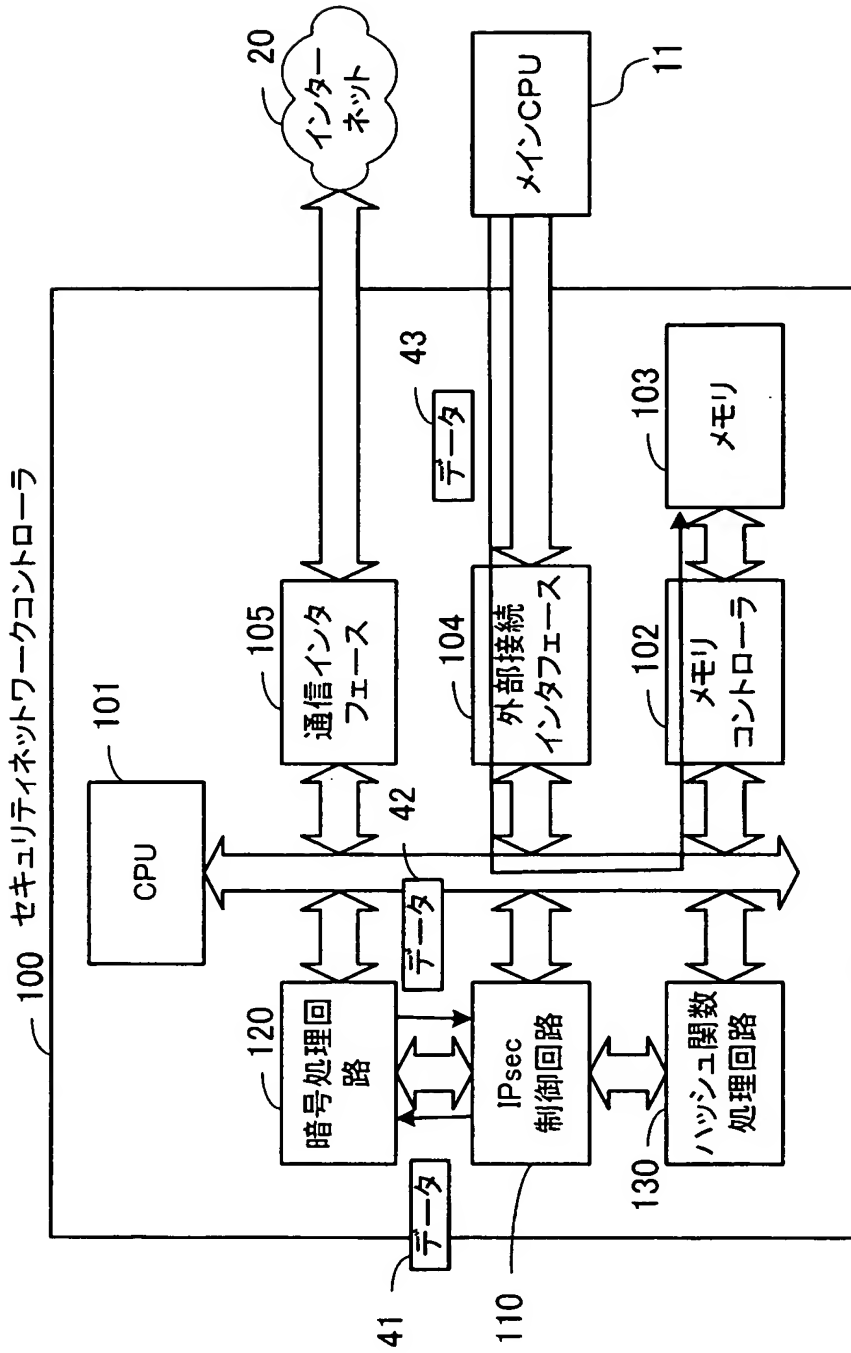
【図5】



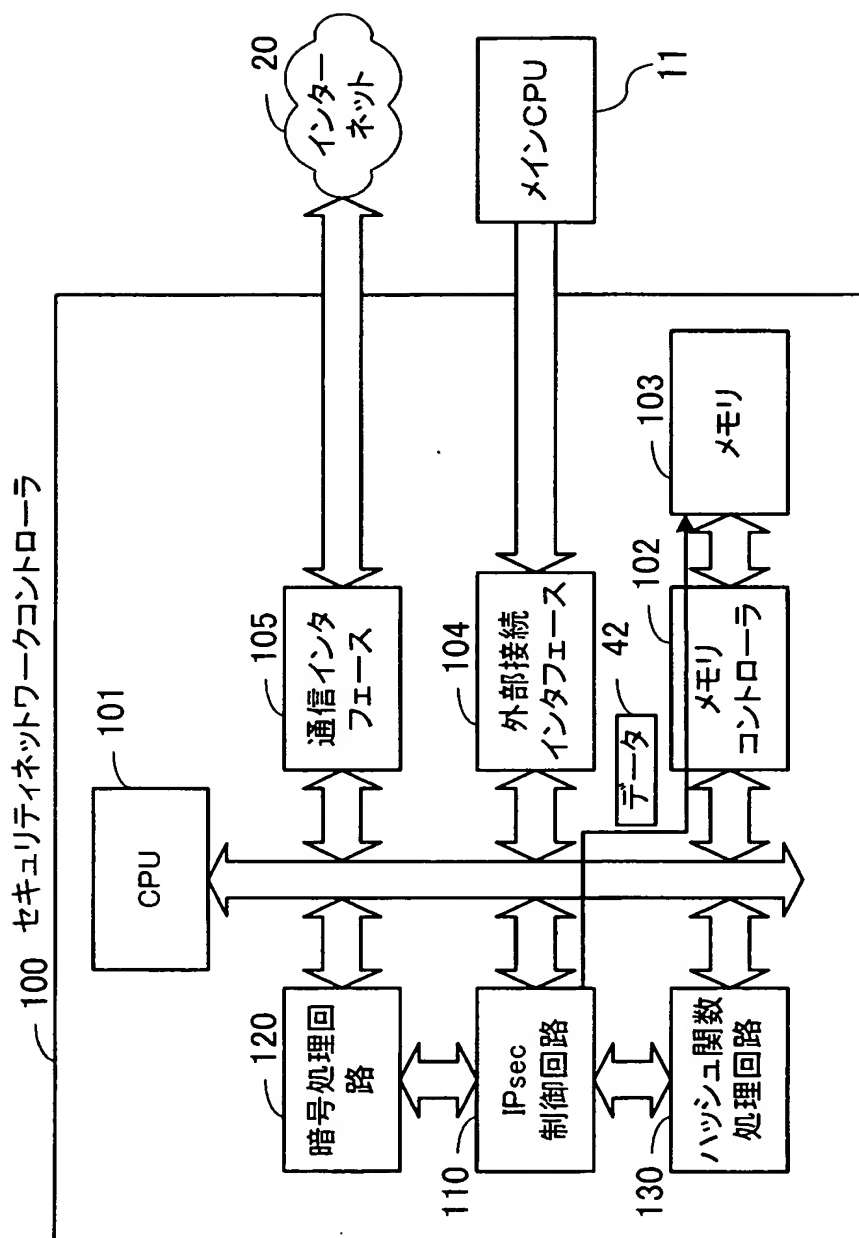
【図 6】



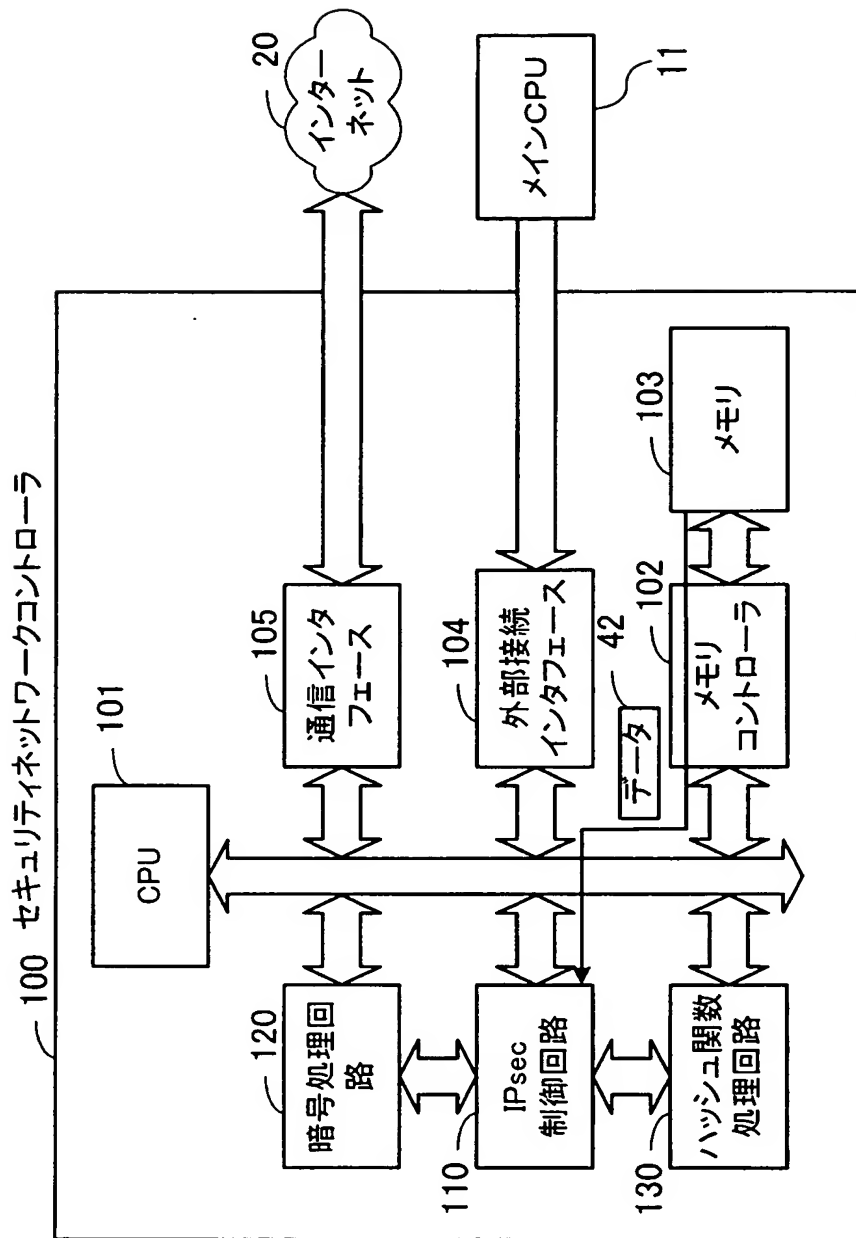
【図 7】



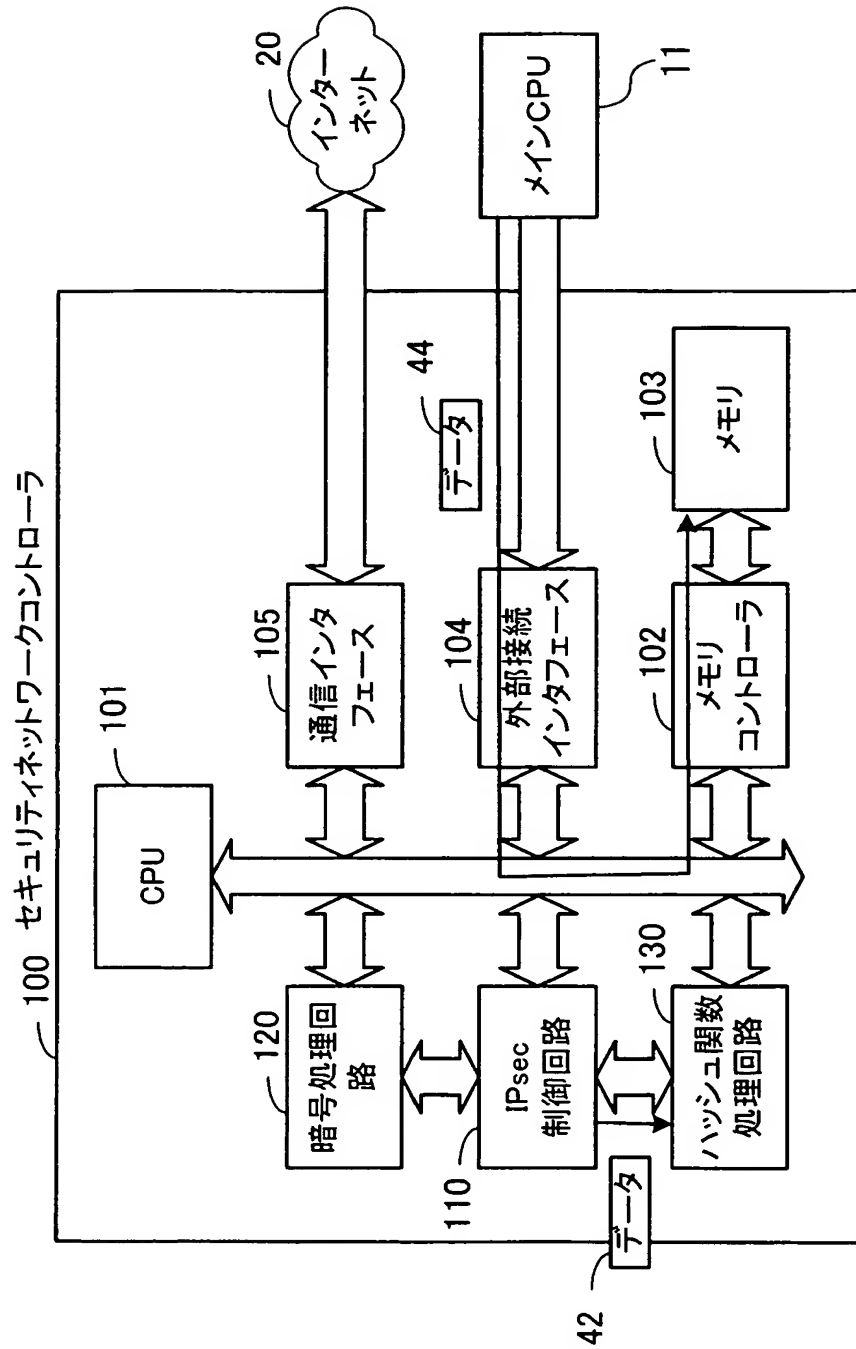
【図8】



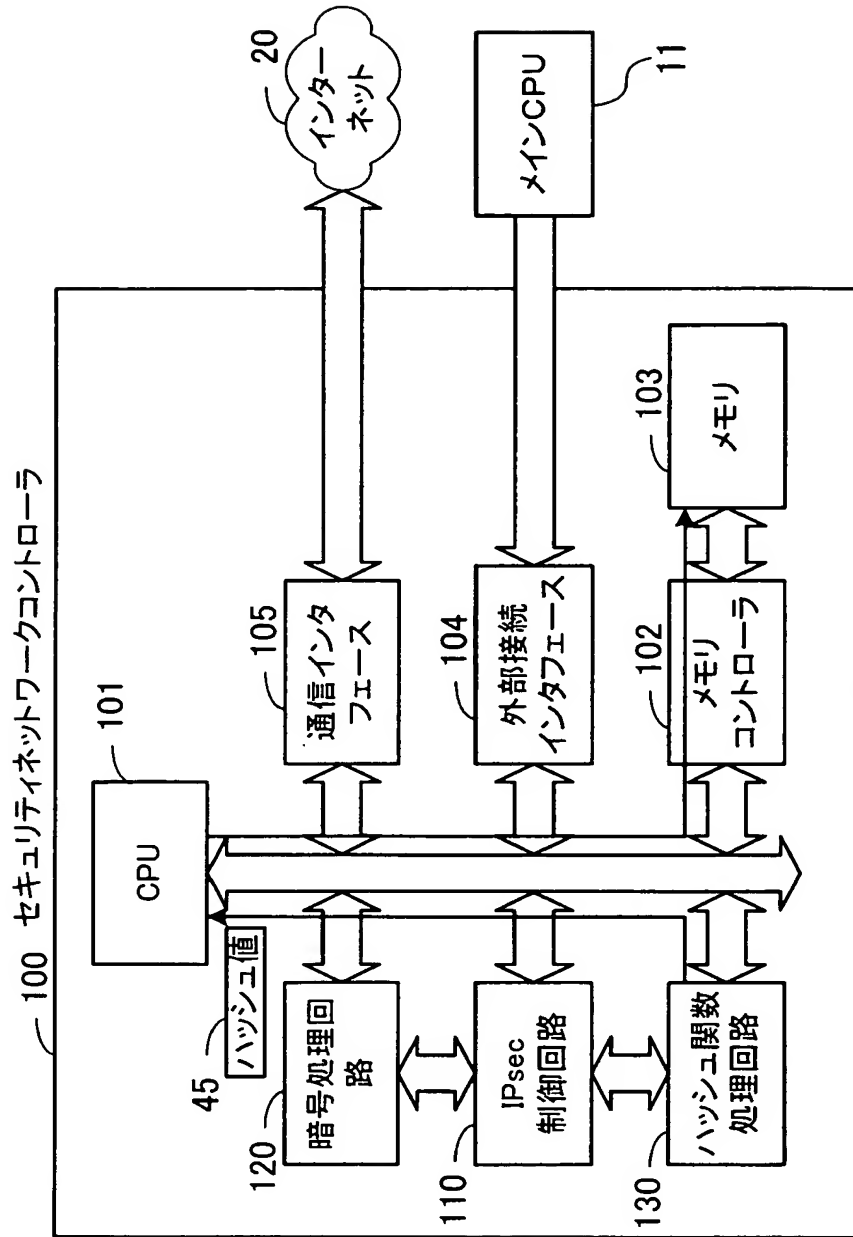
【図 9】



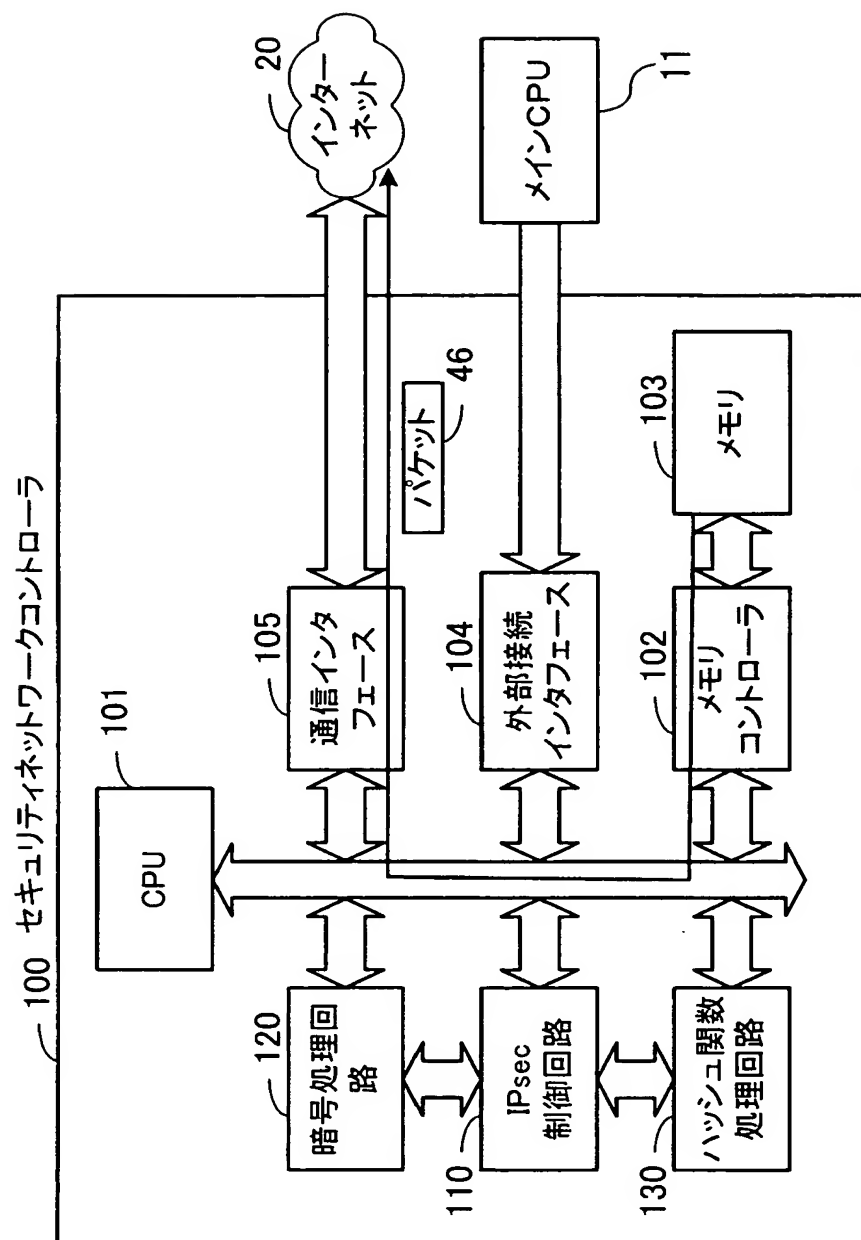
【図10】



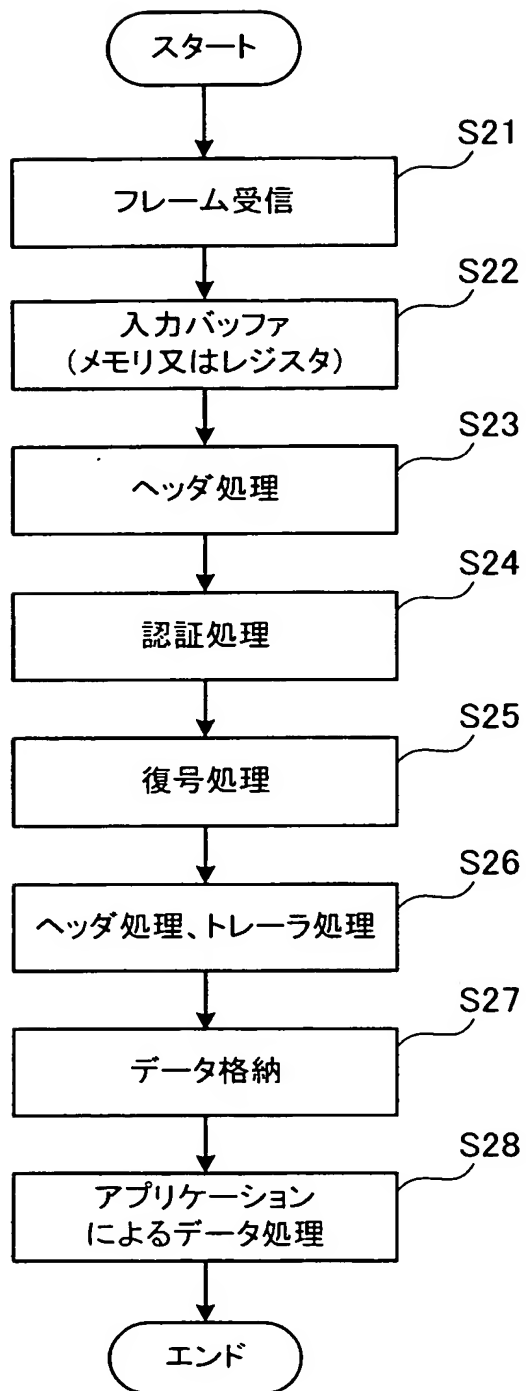
【図 11】



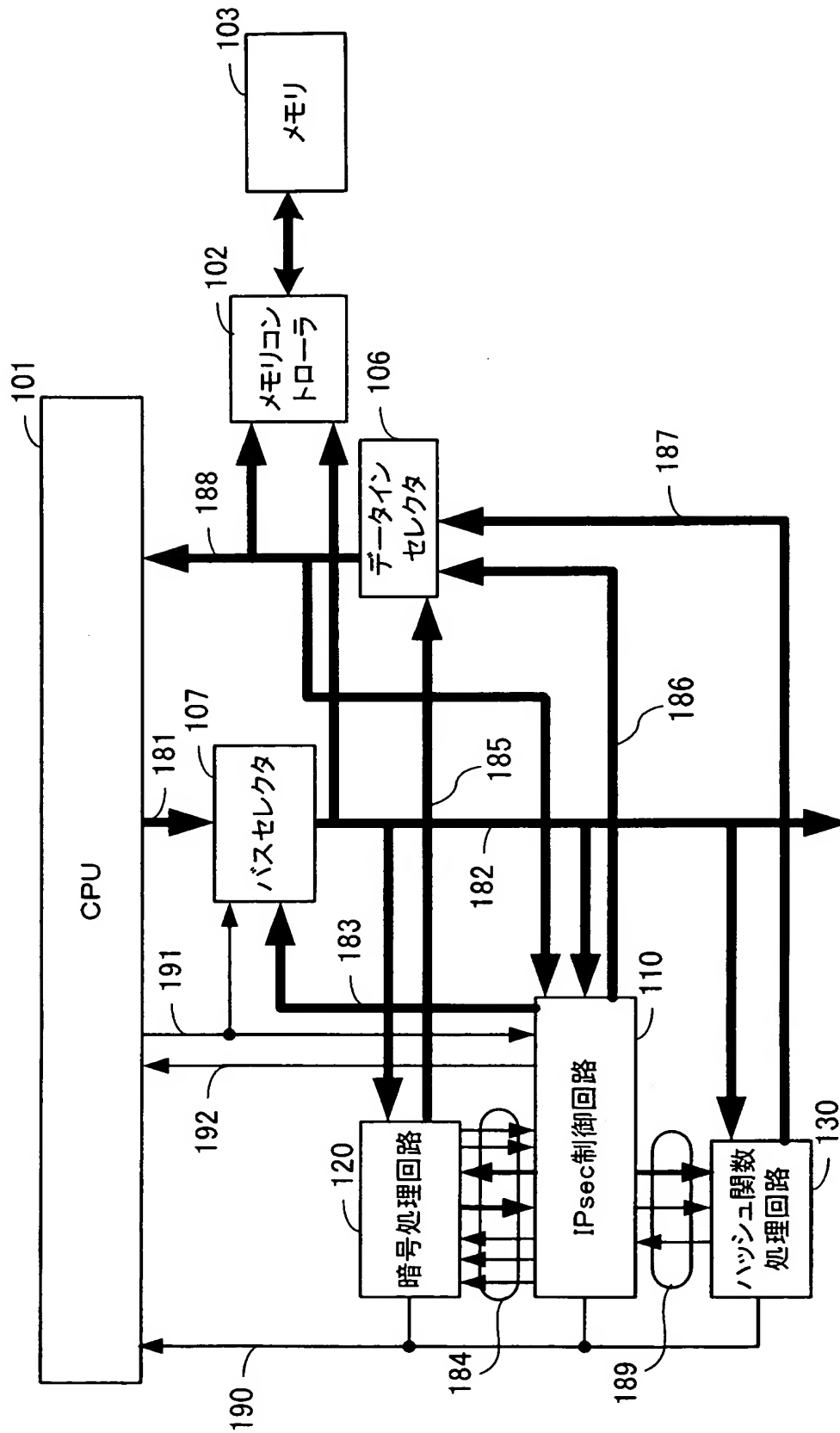
【図 12】



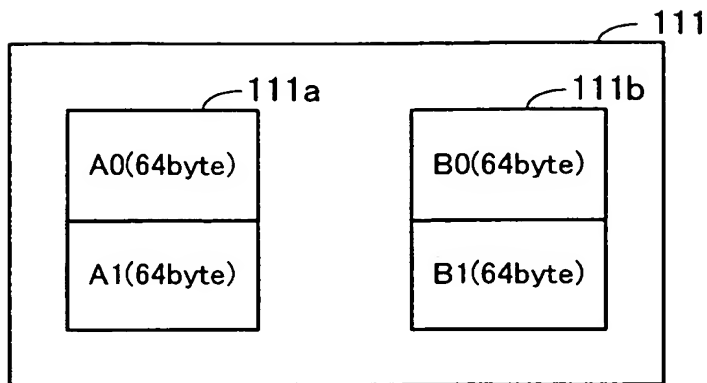
【図 13】



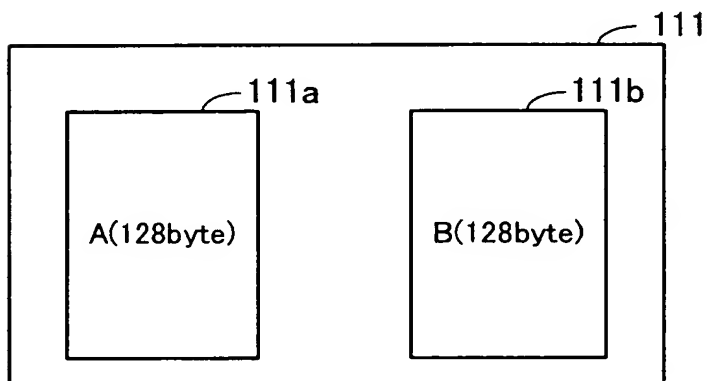
【図 14】



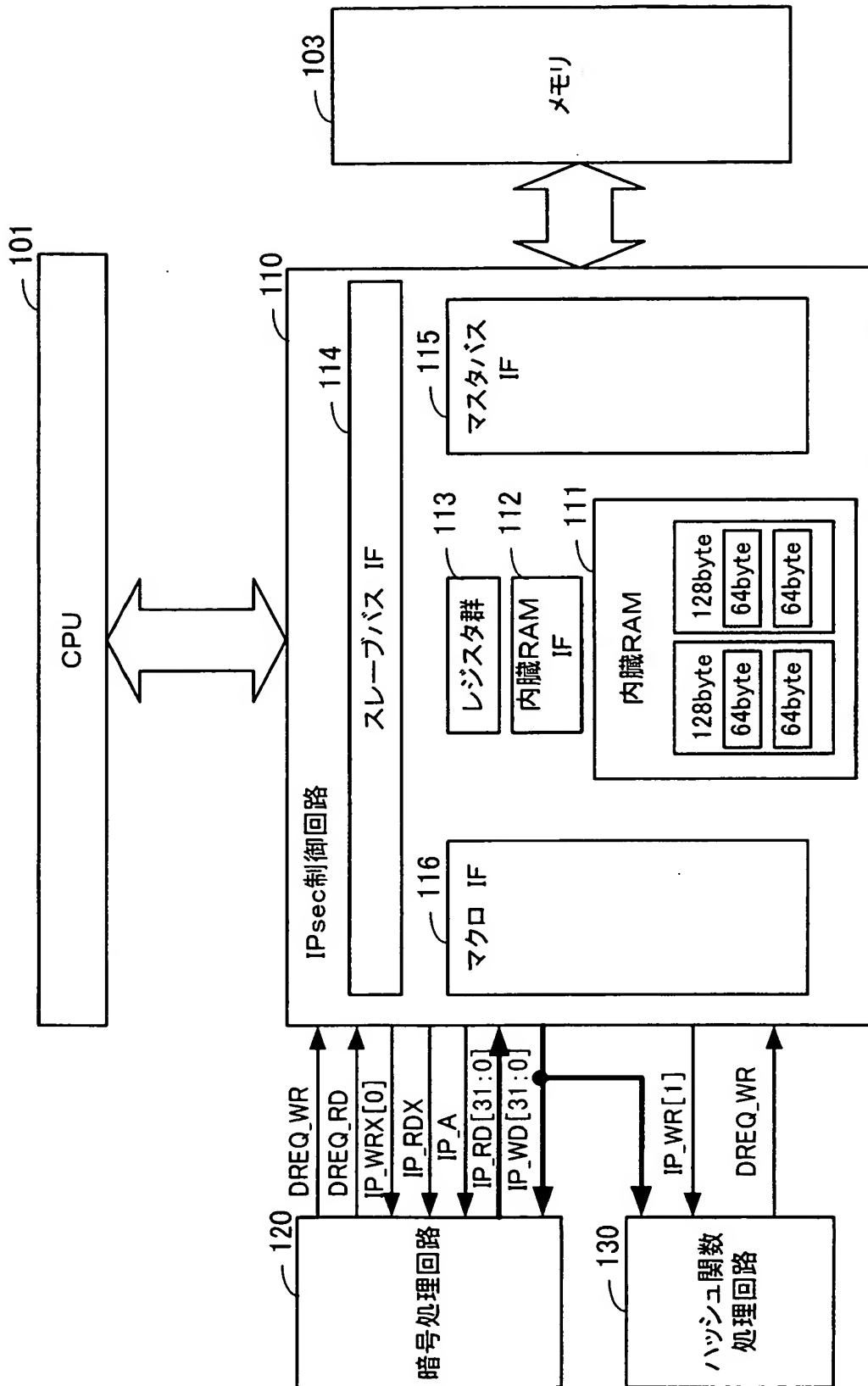
【図 15】



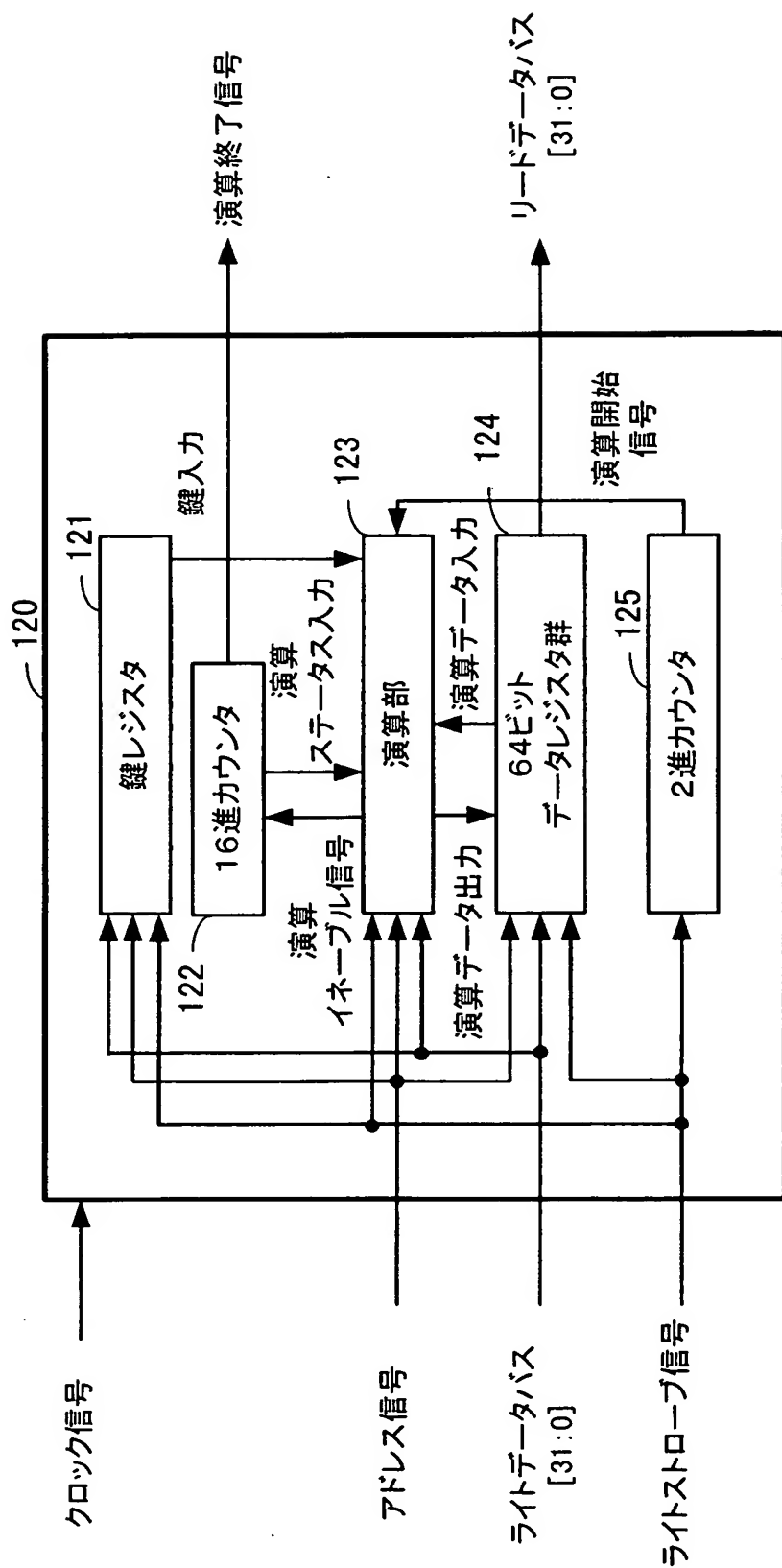
【図 16】



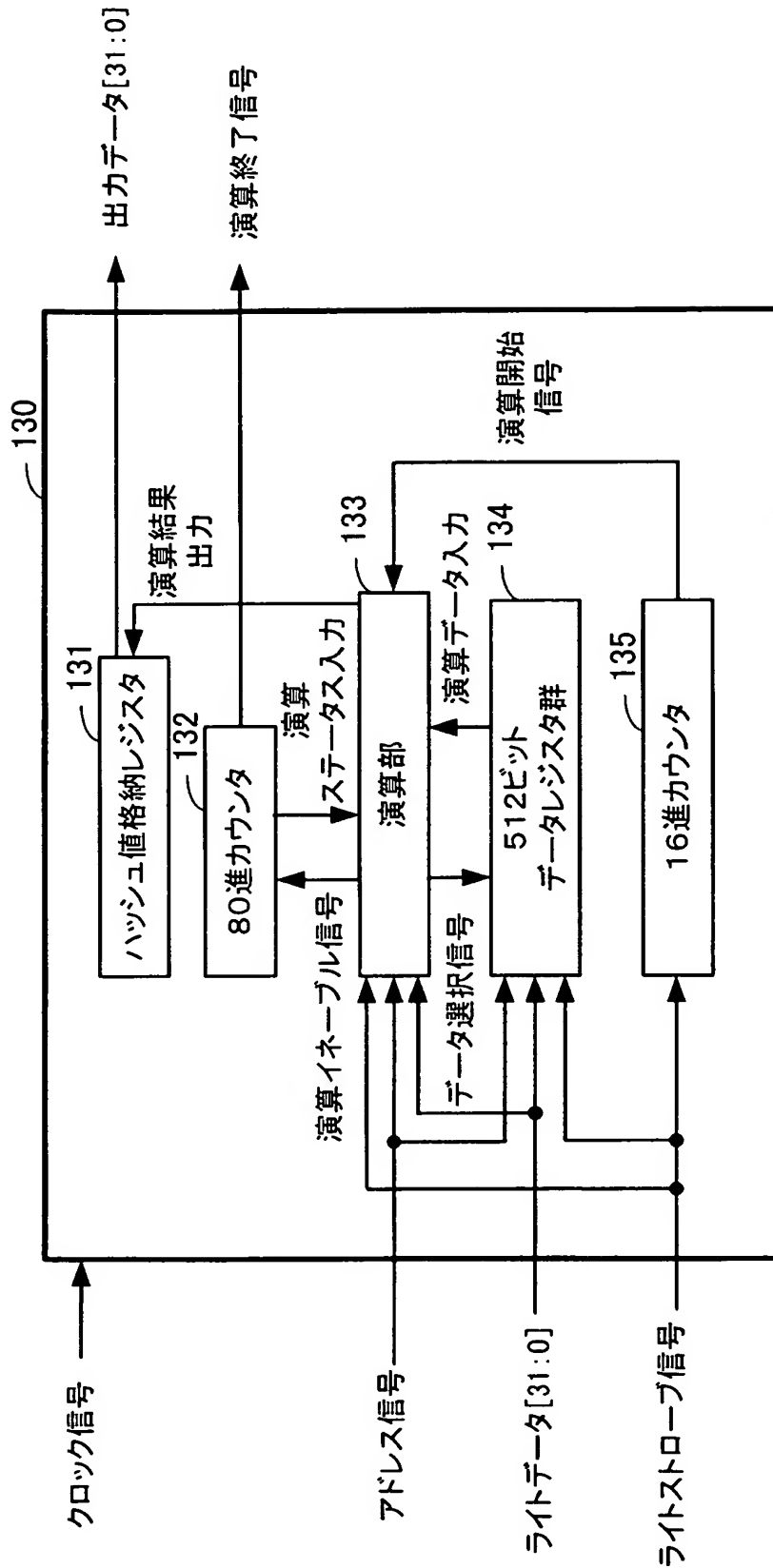
【図 17】



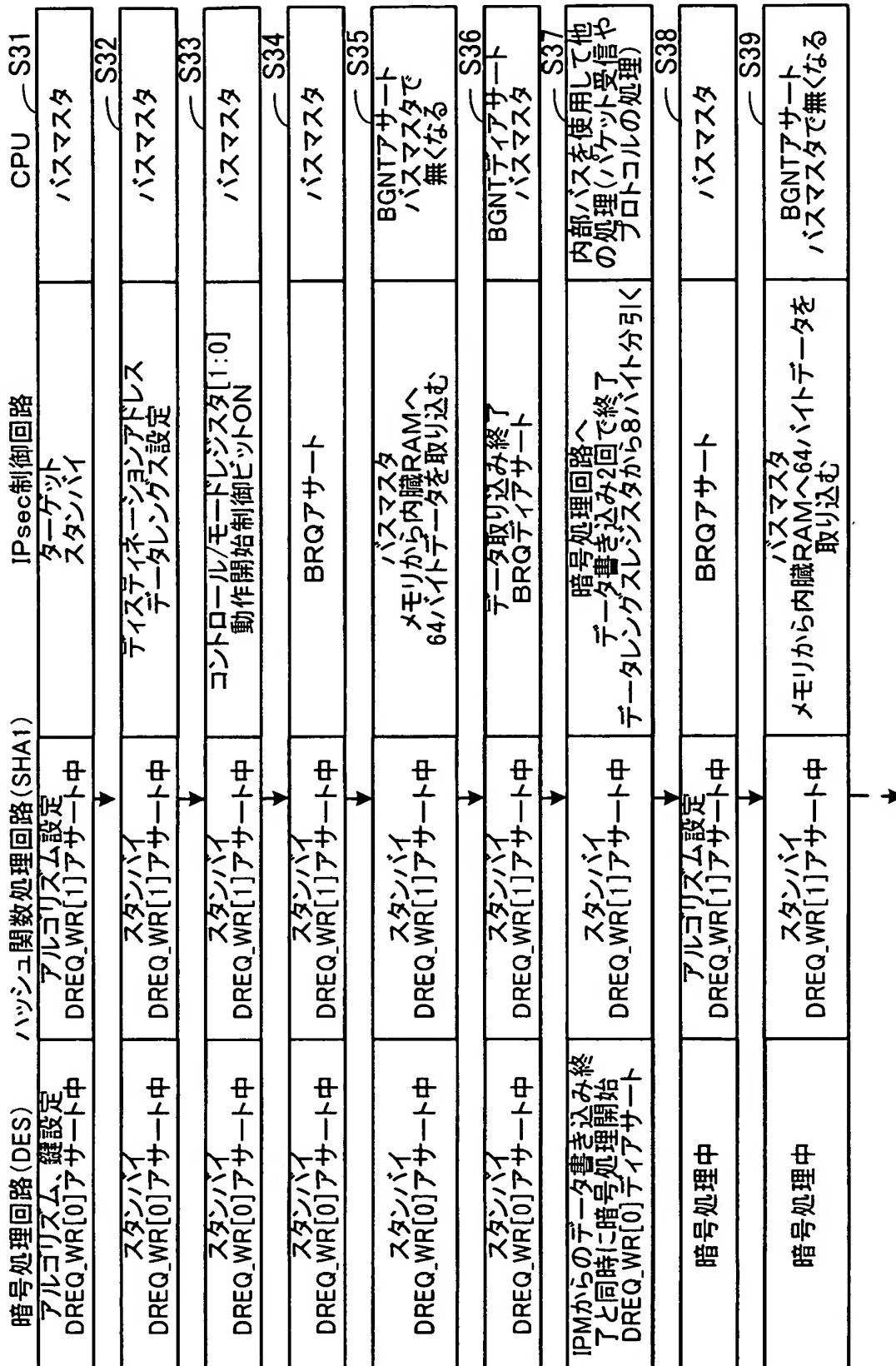
【図 18】



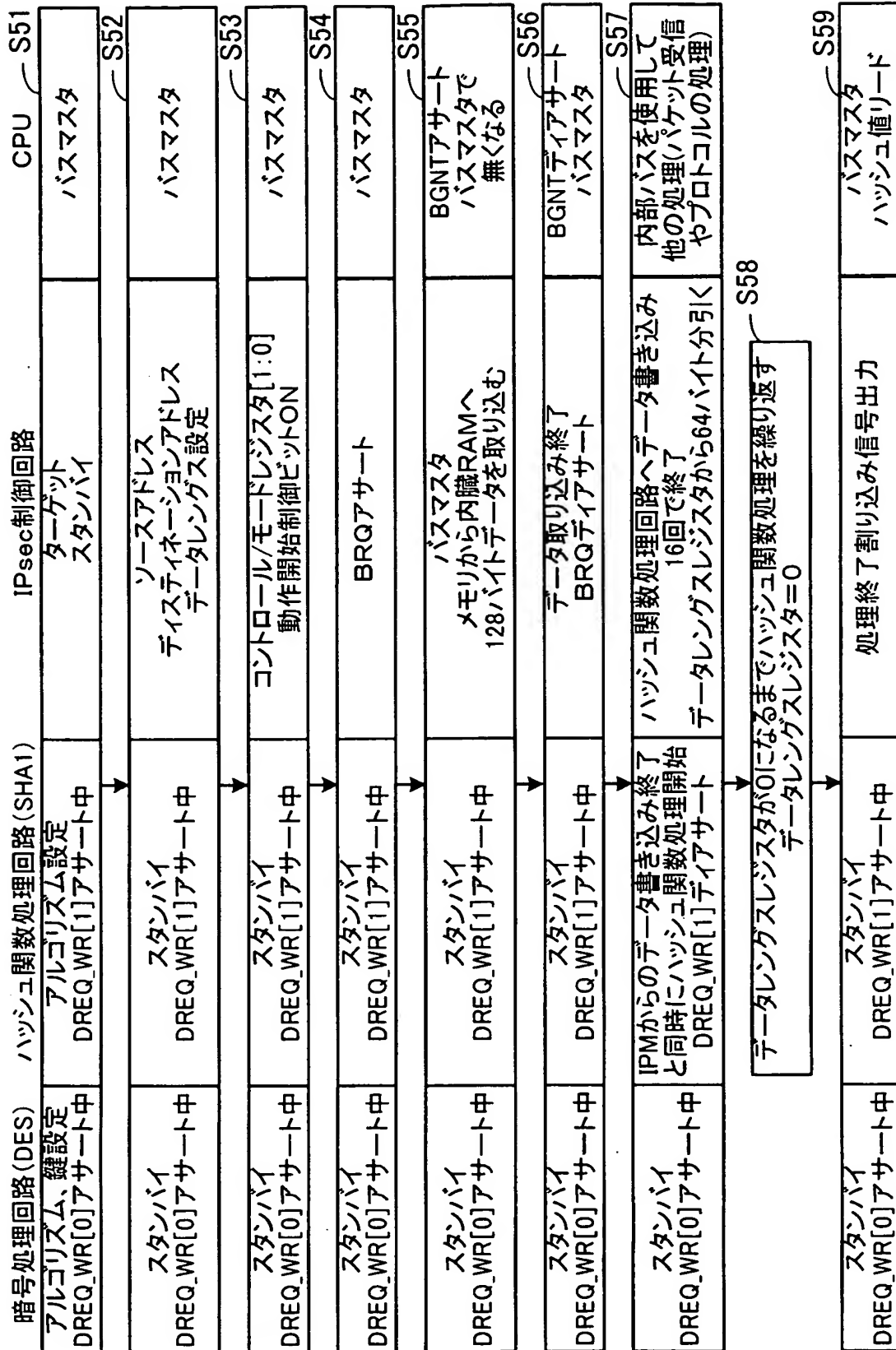
【図 19】



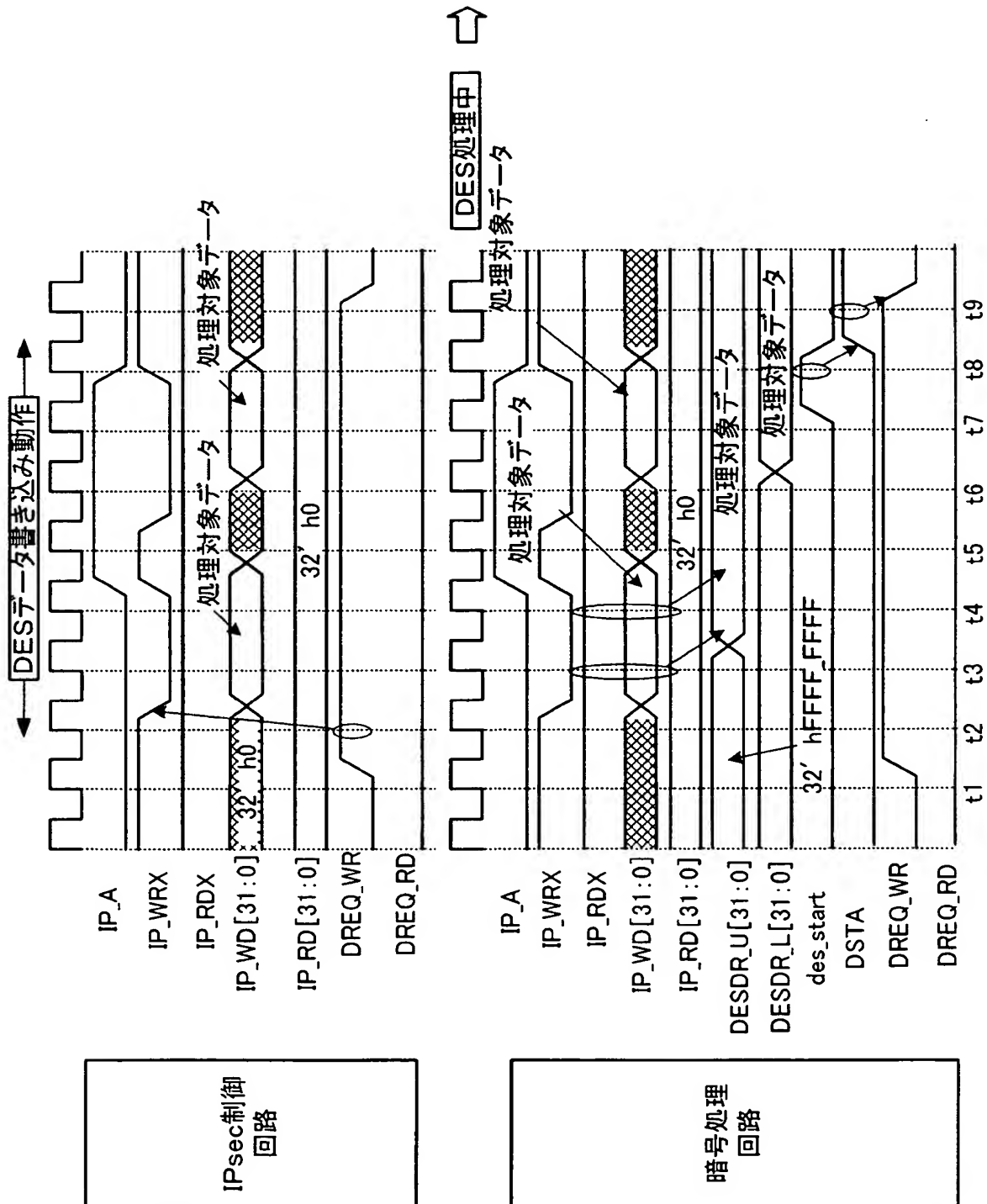
【図 20】



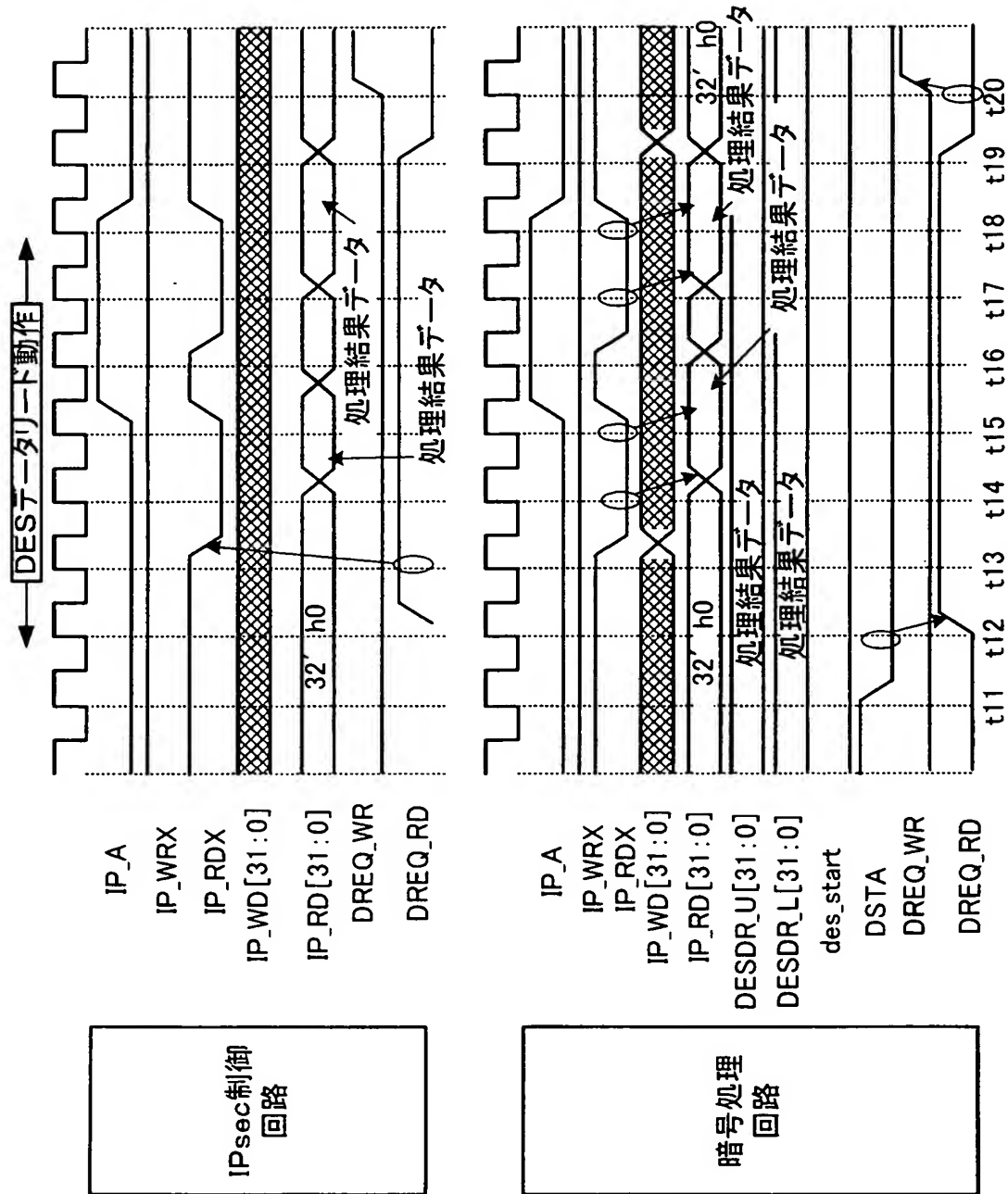
【図 22】



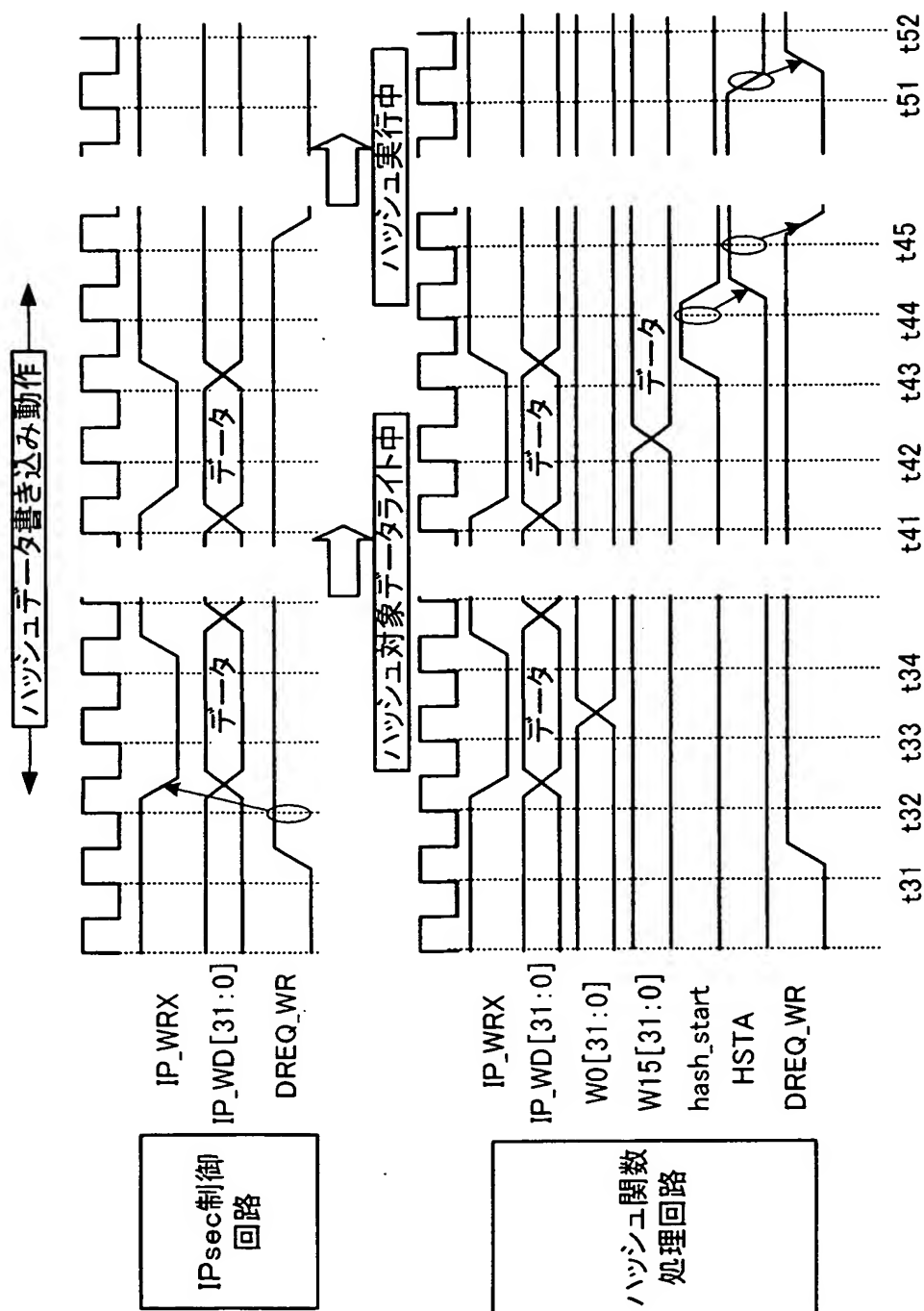
【図 23】



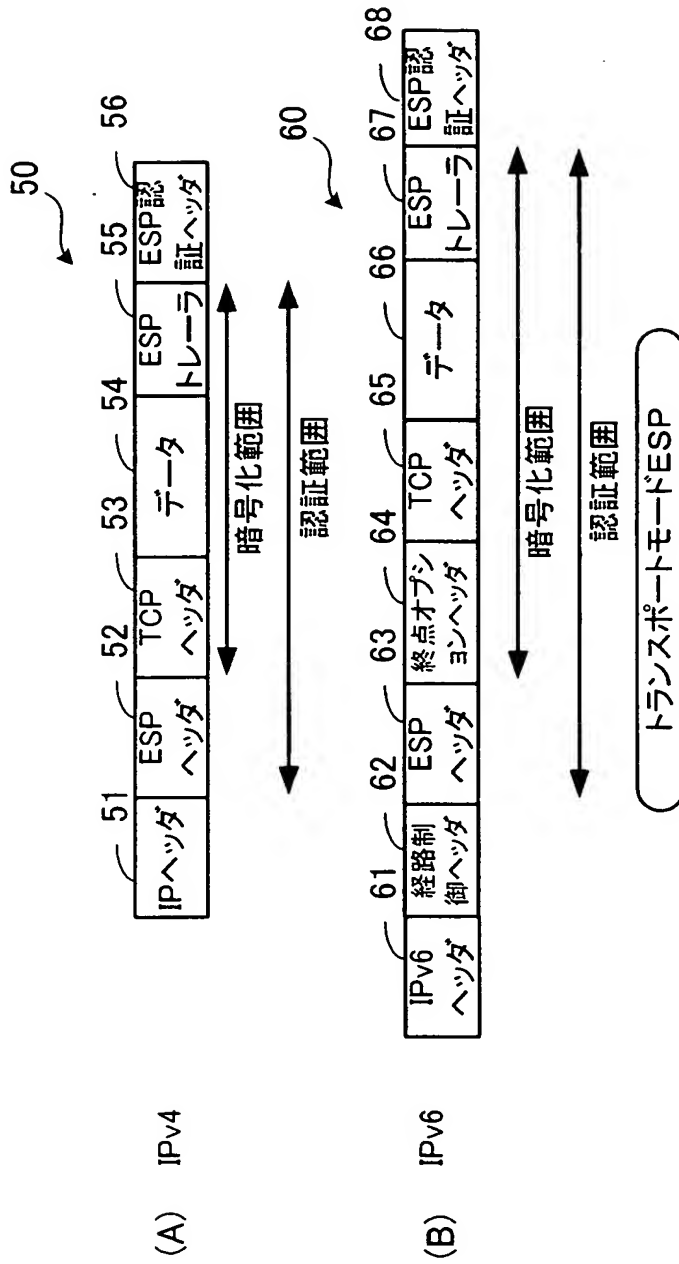
【図 24】



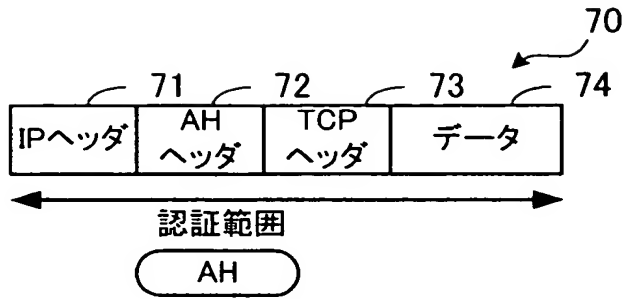
【図 25】



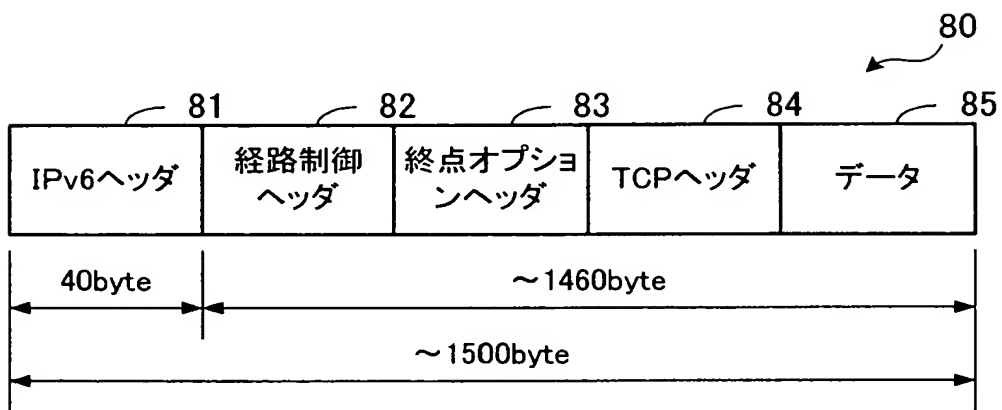
【図 26】



【図 27】

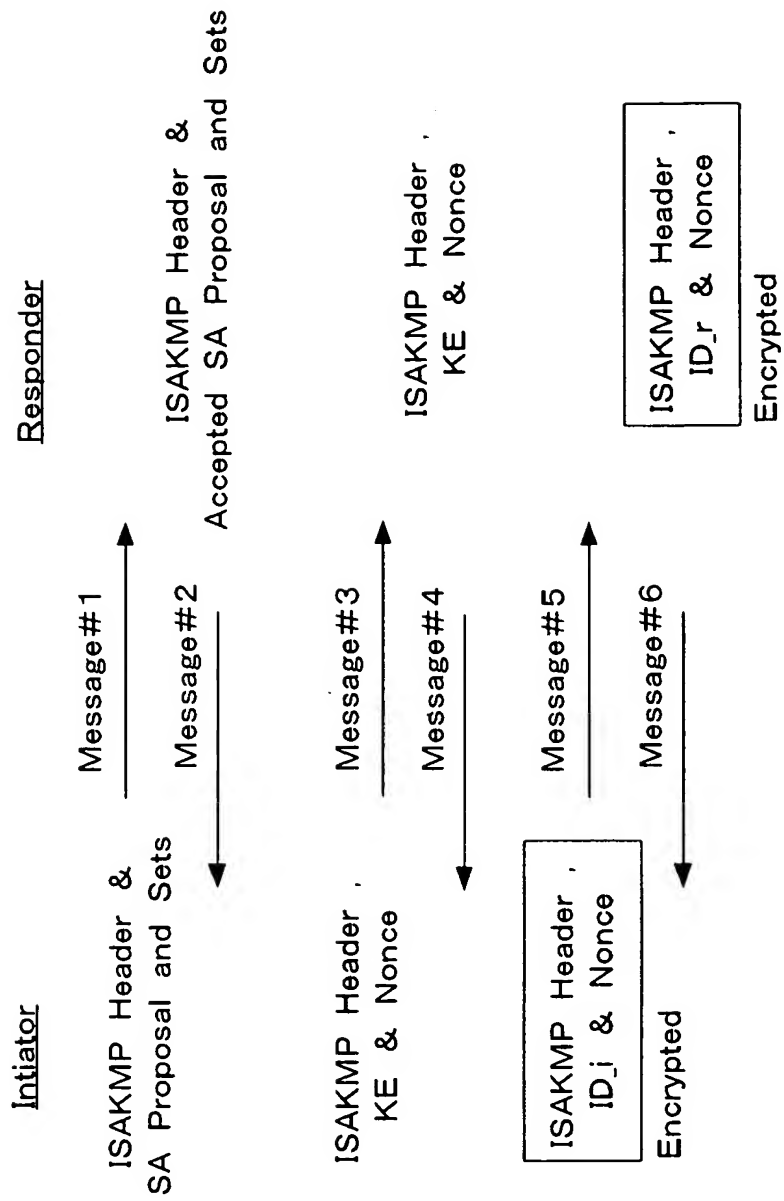


【図 28】



IPパケットサイズ

【図 29】



【図 30】

3DES-CBC-暗号処理 1496バイトデータ処理性能		速度単位は μsec (マイクロ秒)	
		3DES-CBC-暗号化	3DES-CBC-復号化
ソフトウェア		264917	264919
CPU+暗号処理回路		2977	2979
IPsec制御回路+暗号処理回路		579	581

【図 31】

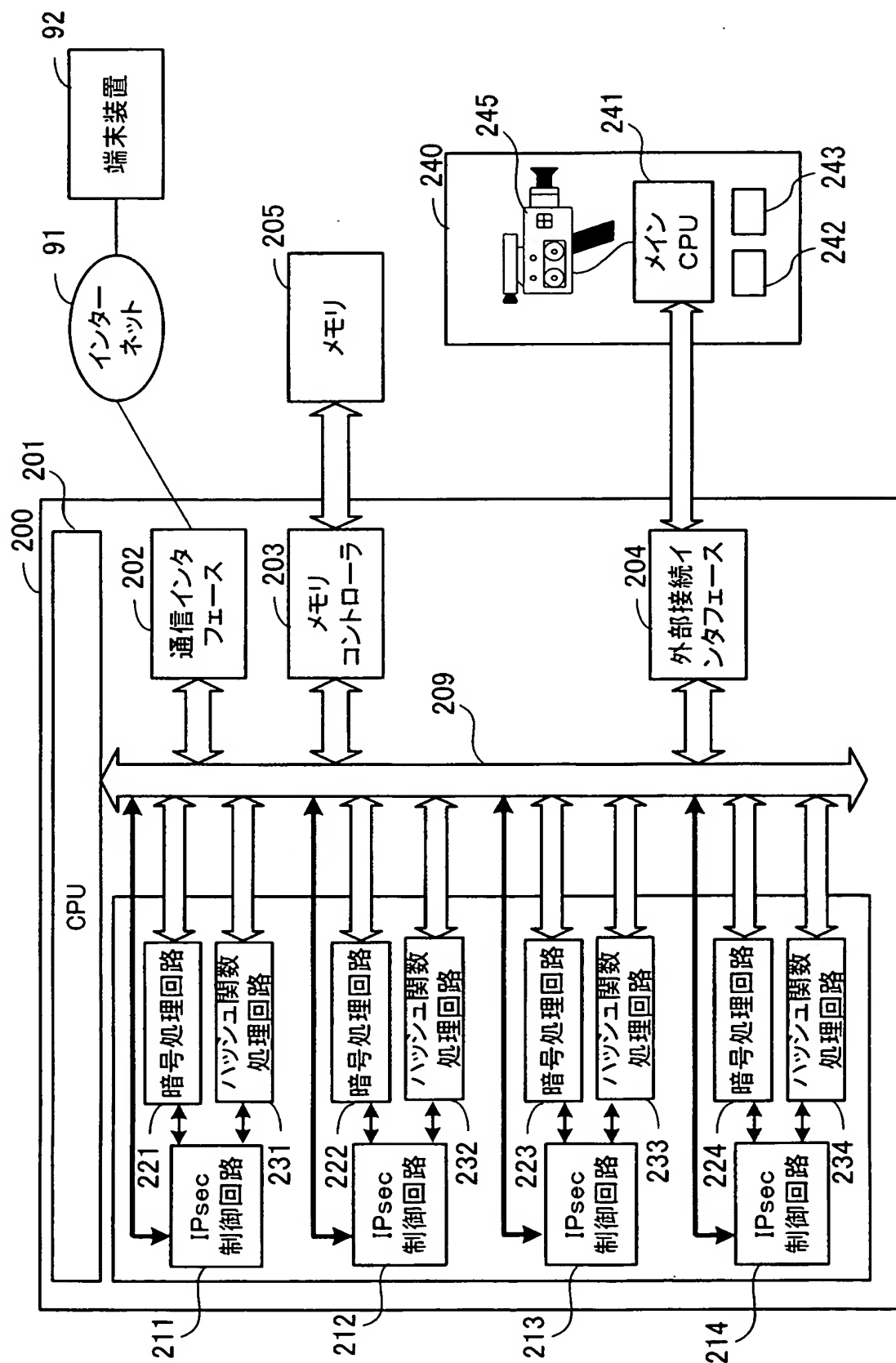
HMAC-SHA1ハッシュ関数処理

1500バイトデータ処理性能

速度単位は μsec (マイクロ秒)

	HMAC-SHA1
ソフトウェア	41309
CPU+ハッシュ関数処理回路	2258
IPsec制御回路+ハッシュ関数処理回路	297

【図 3 2】



【書類名】 要約書

【要約】

【課題】 CPUの処理性能に拘わらず暗号処理を高速に行うことができるようにする。

【解決手段】 データ取得回路1bは、処理対象データ2を取得する。暗号処理回路1cは、入力されたデータの暗号処理を行う。データ入出力制御回路1eは、データ取得回路1bに第1のバス1gを介して接続されると共に暗号処理回路1cに対して第2のバス1hを介して接続されている。データ入出力制御回路1eは、データ取得回路1bが取得した処理対象データ2を第1のバス1g経由で取得して内蔵メモリに格納する。そして、データ入出力制御回路1eは、処理対象データ2を第2のバス1h経由で暗号処理回路1cに入力し、暗号処理回路1cから第2のバス1h経由で暗号処理実行後の処理結果データ3を取得する。

【選択図】 図1

特願 2 0 0 3 - 1 1 2 9 9 2

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 5 2 2 3]

1. 変更年月日

1 9 9 6 年 3 月 2 6 日

[変更理由]

住所変更

住 所

神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

氏 名

富士通株式会社